Clustering Espectral

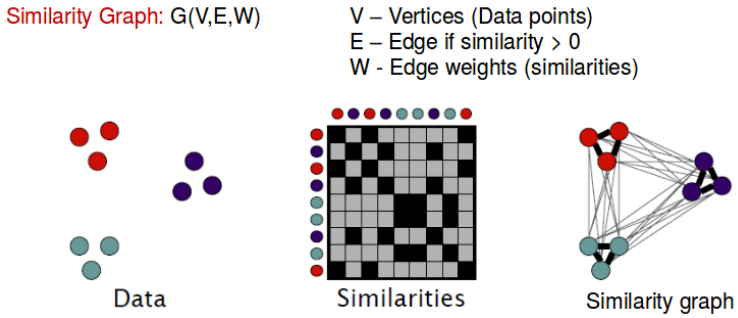
0.1 Introducción

Dado un conjunto de puntos $x_1, x_2, ..., x_n$ y una noción de similitud $s_{ij} \geq 0$ entre todos los pares de puntos x_i y x_j, el objetivo de un algoritmo de clustering es dividir los puntos en varios grupos de manera que puntos en el mismo grupo sean similares y puntos en grupos diferentes no se parezcan.

Dentro de los algoritmos de clustering, una manera de representar el conjunto de datos $\{x_i\}_{i=1}^n$ es mediante un grafo de similitud $G = (V, E)$ donde cada vértice v_i en el grafo representa el punto x_i y donde dos vértices v_i, v_j están conectados si la similitud s_{ij} entre los puntos correspondientes x_i y x_j es positiva o mayor a un cierto umbral. La representación del grafo normalmente se hace mediante una matriz W de tamaño $n \times n$ denominada matriz de adyacencia, donde el elemento w_{ij} representa el peso asociado a la arista que conecta a los vértices v_i y v_j.

Con esto se reformula la tarea de agrupación usando un grafo de adyacencia. Queremos particionar el grafo de forma que las aristas entre diferentes grupos tengas pesos bajos y las aristas dentro de un mismo grupo, pesos altos. La solución óptima para este tipo de problema generalmente es de la clase NP-hard por lo que se han propuesto diferentes algoritmos que den una solución aproximada.

Figure 1: Se representan los datos mediante los vértices de un grafo, cuyas aristas indican la similitud entre pares de vértices.



Los algoritmos de clustering espectral buscan dar una solución aproximada al problema de particionamiento del grafo (y con esto al problema de clustering), haciendo uso de la matriz laplaciana L asociada al grafo que se quiere particionar y la cual se puede construir a partir de la matriz W que ya se tiene.

La manera en que se usa la matriz laplaciana es obteniendo sus k vectores propios asociados a los valores propios más pequeños, los cuales nos dan cierta información sobre el grafo,

que le será útil a los algoitmos de clusterización clásicos como k-means para realizar la agrupación de los datos.

0.2 Metología

Como ya se comentó, el algoritmo parte de un conjunto de datos $\{x_i\}_{i=1}^n$ con $x_i \epsilon\ \mathbb{R}^p$. A partir de dicho conjunto de datos se construye la matriz de adyacencia ponderada $W \epsilon\ \mathbb{R}^{n \times n}$, con $W_{ij} = s(i,j)$. Donde $s(i,j)$ es una función de similitud, que como su nombre lo dice, trata de dar una medida de qué tan cercanos o parecidos son los puntos x_i y x_j. Para este proyecto se ha usado la función de similitud $s(i,j) = exp(\|x_i - x_j^2\|/(2\sigma)^2)$.

La matriz diagonal de grado $D \epsilon\ \mathbb{R}^{n \times n}$, se construye a partir de la matriz W y tiene los siguientes valores:

$$D_{ij} = \begin{cases} \sum_{k=1}^{n} W_i k & i = j \\ 0 & i \neq j \end{cases}$$

Para construir la matriz laplaciana, se usa la siguiente ecuación:

$$L = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{1}$$

Donde I_n es la matriz identidad de tamaño $n \times n$. Es conocido que $L \epsilon\ \mathbb{R}^{n \times n}$ es una matriz simétrica y positiva semidefinida, por lo que tiene valores porpios reales no negativos $0 \leq \lambda_i$ para $i = 1, ..., n$.

Un vez obtenida la matriz laplaciana L, se calculan sus k vectores propios y sus k valores propios más pequeños. Para esto, se ha usado el eigensolver de Jacobi ya que es el que mejores resultados ha dado.

Una vez que se tienen identificados los k vectores propios asociados a los k valores propios más pequeños, se forma la matriz $V \epsilon\ \mathbb{R}^{n \times k}$, cuyas columnas son los vectores propios calculados. De dicha matriz se toman sus renglones para formar nuevos datos $y_i \epsilon\ \mathbb{R}^k$ con $i = 1, ..., n$.

Finalmente, se aplica el algoritmo de clustering denominado k-means sobre el conjunto de puntos $\{y_i\}_{i=1}^n$. En dicho algoritmo de clustering se define el número de clusters c en los que se desea particionar el conjunto $\{y_i\}_{i=1}^n$. Dichos clusters estarán identificados mediante el conjunto $C = \{0, 1, ..., c\}$, con lo que cada punto y_i estará asociado a un sólo elemento de C que también corresponderá al cluster que se le asignará al punto original x_i.

Proyecto Final

0.3 Algoritmo

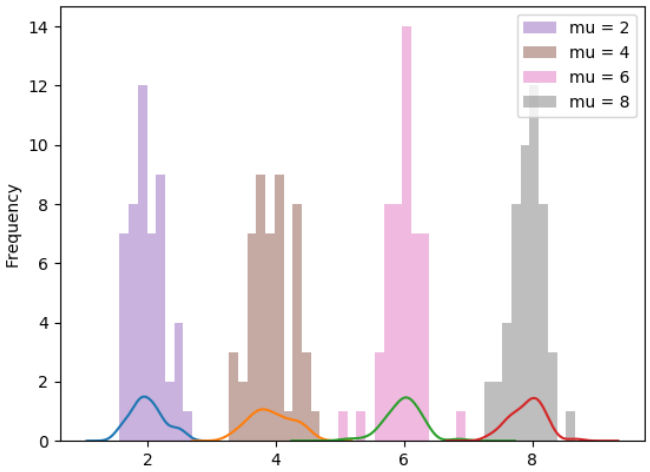
A manera de resumen, el algoritmo de clustering espectral queda de la siguiente forma:

1. Construir matrices W, D y L.
2. Obtener los k vectores propios $v_1, v_2, ..., v_k$ de L asociados a los valores propios $\lambda_1, ..., \lambda_k$ mś pequeños.
3. Construir matriz $V \epsilon \mathbb{R}^{n \times k}$ cuyas columnas son los k vectores propios calculados.
4. Iterpretar los renglones $y_i \epsilon \mathbb{R}^k$ de V como los nuevos datos con $i = 1, 2, ..., n$.
5. Aplicar algoritmo de agrupación de datos en los puntos y_i.

0.4 Resultados

Para poder verificar el funcionamiento del algoritmo, se creó un dataset $\{x_i\}_{i=1}^{200}$ compuesto por datos provenientes de 4 distribuciones normales con medias $\mu_1 = 2$, $\mu_2 = 4$, $\mu_3 = 6$ y $\mu_4 = 8$, todas con varianza $\sigma = 0.3$. Se generaron 50 datos de cada distribución, teniendo así, en total, 200 datos. En la siguiente imagen se muestra un histograma de dichos datos.

Figure 2: Dataset de 4 distribuciones normales con $n = 200$.



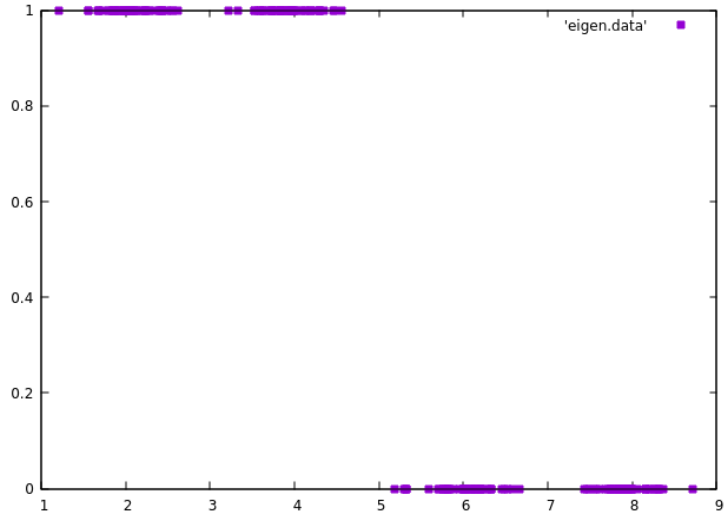
Para verificar que los vectores propios contienen información acerca del agrupamiento de los datos, se grafica el valor de cada dato x_i en ej eje x contra la componente i-ésima v_i del vector propio v, ya que ambos son de dimensión 200. Para visualizar de manera más clara el

tipo de información que contienen las componentes del vector propio, en vez de gráficar las componentes como tal, se transformarán a 0 si el valor $v_i < 0$ o a 1 si $v_i \geq 0$. Es decir:

$$v_i = \begin{cases} 0 & v_i < 0 \\ 1 & v_i \geq 0 \end{cases}$$

Para el primer vector propio, es decir, el vector propio asociado al valor propio más pequeño, se obtiene la siguiente gráfica:

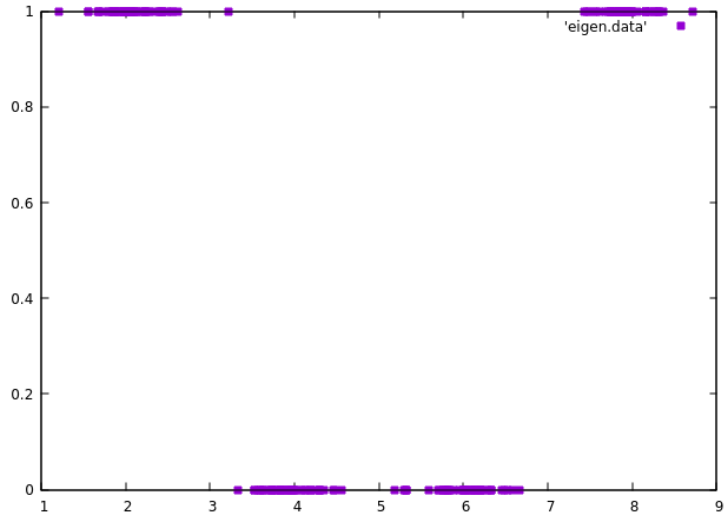
Figure 3: Gráfica v_i vs x_i para primer vector propio v.



Se puede observar que agrupa los datos que provienen de las distribuciones normales con medias $\mu = 2$ y $\mu = 4$ en un mismo cluster y los datos que provienen de las distribuciones normales con medias $\mu = 6$ y $\mu = 8$ en otro cluster.

Para el segundo vector propio, es decir, el vector propio asociado al segundo valor propio más pequeño, se obtiene la siguiente gráfica:

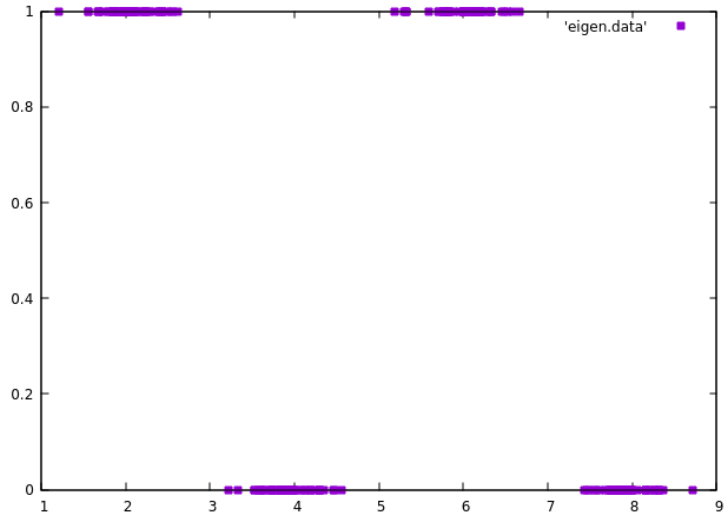
Figure 4: Gráfica v_i vs x_i para segundo vector propio v.



Se puede observar que agrupa los datos que provienen de las distribuciones normales con medias $\mu = 2$ y $\mu = 8$ en un mismo cluster y los datos que provienen de las distribuciones normales con medias $\mu = 4$ y $\mu = 6$ en otro cluster.

Para el tercer vector propio, es decir, el vector propio asociado al tercer valor propio más pequeño, se obtiene la siguiente gráfica:

Figure 5: Gráfica v_i vs x_i para tercer vector propio v.



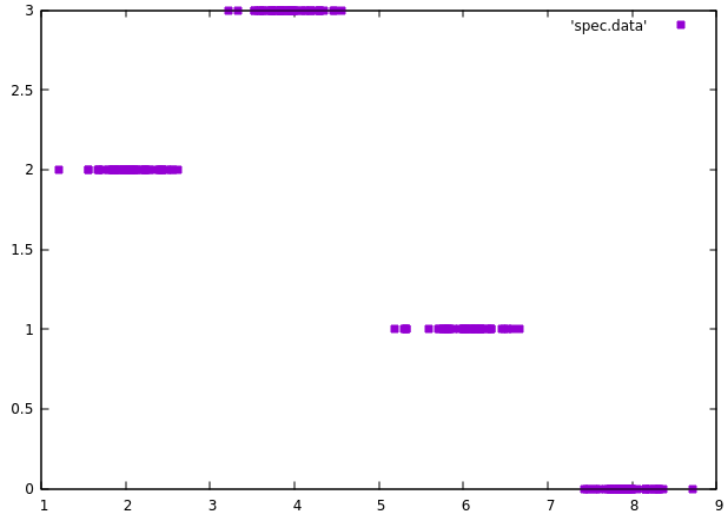
Se puede observar que agrupa los datos que provienen de las distribuciones normales con medias $\mu = 2$ y $\mu = 6$ en un mismo cluster y los datos que provienen de las distribuciones normales con medias $\mu = 4$ y $\mu = 8$ en otro cluster.

Proyecto Final

Finalmemte, se muestra el resultado del agrupamiento de los datos en 4 clusters, ya que originalmente se tienen 4 grupos (4 distribuciones normales). El algoritmo de clustering le asigna a cada punto x_i del dataset, una etiqueta con valores $\{0, 1, 2, 3\}$, los cuales representan el identificador del cluster al que pertenece el punto x_i.

El algoritmo se ejecuta usando los primeros 3 vectores propios de la matriz laplaciana L, es decir, se realiza el agrupamiento de los datos $\{x_i\}_{i=1}^{200} \epsilon\ \mathbb{R}$ con puntos $\{y_i\}_{i=1}^{200} \epsilon\ \mathbb{R}^3$.

Figure 6: Resultado de clustering espectral para dataset de 4 distribuciones normales. El eje x representa el valor de los puntos x_i y el eje y el cluster que le asignó el algoritmo.



Como se puede observar, se ha logrado agrupar todos los datos correctamente, asignando el cluster de acuerdo a la distribución normal de la que proviene cada dato.

0.5 Conclusiones

Para la implementación de este algoritmos hay varios parámetros que se deben tomar en cuenta para lograr un correcto funcionamiento, ya que dependiento de los datos que se quieran agrupar, funcionará mejor una configuración u otra:

- La función de similitud s_{ij} a usar para construir la matriz de adyacencia W.
- La variante de matriz laplaciana L que se vaya a usar
- El algoritmo a usar para la obtención de los vectores y valores propios de la matriz L.
- El algoritmos de clustering para agrupar los puntos y_i.

Por lo que es bastante importante conocer cada componente del algoritmo y como afectan en el resultado final del agrupamiento de los datos.

Por otro lado, también se destaca la importacia de los métodos numéricos, en este caso, el cálculo de vectores y valores propios de una matriz. Es bastante importante tener un algoritmo eficiente y que pueda manejar bien los errores numércios ya que en muchas aplicaciones el resultado dependerá de qué tan buenos sean los cálculos de método numérico que se esté usando.

Finalmente, se hace la observación de que a pesar de haber dado buenos resultados en el dataset creado, aún faltaría probar con datasets más complejos o incluso con imágenes. También faltaría explorar diferentes funciones de similitud, vaiantes de la matriz laplaciana y aplicar otros algoritmos de clustering para determinar cuales son las mejores opciones.