Clusters Espectrales

Nicolás Kossacoff

Noviembre 2024

1. Grafos

En esta primera sección vamos a presentar algunos de los conceptos matemáticos más utilizados para clusters espectrales.

1.1. Grafos de similaridad

Nuestro objetivo es poder agrupar nuestro conjunto de datos, $X = \{x_1, \ldots, x_n\}$, en grupos tales que las observaciones dentro de un mismo grupo sean lo más similares entre sí y las observaciones que se encuentran en grupos distintos sean disimilares entre sí.

Si las similaridades entre observaciones es la única información con la que contamos, podemos representar nuestros datos en forma de un **grafo de similaridades**, $G = (V, E)$, donde V es el vector de **vértices** del grafo (cada vértice representa una observación en X) y E representa las **aristas** (i.e., conjunto no ordenado de pares de observaciones). Dos vértices están conectados si la similaridad entre la observación i y la observación j, $s_{i,j}$, es positiva o mayor a un umbral.

Decimos que un grafo es **pesado** si cada arista entre dos vértices i y j tiene asignada un peso $w_{i,j} \geq 0$. Notar que si $w_{i,j} = 0$ entonces esos vértices no se encuentran conectados.

Dicho esto, nuestro problema de clustering se puede reformular de la siguiente manera: queremos encontrar una partición de nuestro grafo tal que los vértices en grupos diferentes tengan pesos bajos (i.e., sean disimilares entre sí) y los vértices de un mismo grupo tengan pesos altos (i.e., sean similares entre sí).

1.1.1. Notación

Sea G un grafo **no dirigido** con vértices $V = \{v_1, \ldots, v_n\}$. Un grafo es no dirigido si las relaciones entre los vértices son simétricas.

Definimos la **matriz de adyacencia**, la cual contiene los pesos asociados a cada

arista:

$$W = \begin{pmatrix} w_{1,1} & \dots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \dots & w_{n,n} \end{pmatrix}$$

Definimos también el **grado** de un vértice $i \in V$ como:

$$d_i = \sum_{j=1}^{n} w_{i,j}$$

y a la **matriz de grados** como la matriz diagonal:

$$D = \text{diag}(d_1, \dots, d_n)$$

Sea A un sub-conjunto de los vértices, $A \subset V$. Luego, para dos conjuntos no necesariamente disjuntos, $A, B \subset V$, definimos:

$$W(A, B) = \sum_{i \in A, j \in B} w_{i,j}$$

Definimos también el vector indicador $\mathbb{1}_A = (f_1, \dots, f_n)' \in \mathbb{R}^n$, el cual toma el valor $f_i = 1$ si el vértice $i \in A$ y toma el valor $f_i = 0$ en caso contrario.

Hay dos maneras de medir el tamaño de un sub-conjunto $A \subset V$:

- Podemos utilizar el número de vértices en el sub-conjunto A, $|A|$.
- Podemos calcular el volumen de A, que es equivalente a la suma de los grados de los vértices del sub-conjunto:

$$\text{vol}(A) = \sum_{i \in A} d_i$$

Decimos que un sub-conjunto $A \subset V$ es **conexo** si podemos unir cualquier par de vértices a través de un camino. Esto quiere decir que si tenemos dos vértices $a, b \in A$, entonces existe una sucesión de vértices $\{v_1, \dots, v_m\}$, con $v_1 = a$ y $v_m = b$, tal que $w_{i,i+1} > 0$.

Finalmente, decimos que $A \subset V$ es una **componente conexa** si no existe conexión entre los vértices de A y su complemento $\bar{A}$. Esta claro que las componentes conexas de un grafo forman una partición.

1.1.2. Tipos de grafos de similaridad

Hay muchas maneras de transformar nuestro conjuntos de datos X, con sus respectivas similaridades o distancias, en un grafo (matriz) de similaridad. Lo importante es recordar que, al construir los grafos, lo que buscamos es poder representar lo mejor posible el entorno local de las observaciones.

- **ε-neighborhood graph.** Conecta los vértices cuya distancia es menor a ε. Como las distancias entre todos los vértices que se encuentran conectados son muy parecidas (como máximos son iguales a ε), agregar pesos no es necesario, ya que no aportaría más información.
- **k-nearest neighbor graph.** Conecta al vértice i con el vértice j si este último es uno de los k vecinos más cercanos del primero. Sin embargo, este enfoque tiene un problema, y es que obtenemos grafos dirigidos, ya que la relación no es simétrica[1].

  Hay dos maneras de solucionar este problema y obtener grafos no dirigidos:

  - La primer manera es ignorando la dirección. Esto quiere decir que dos vectores $i, j \in V$ van a estar conectados si i se encuentra dentro de los k vecinos más cercanos de j **o** j se encuentra dentro de los k vecinos más cercanos de i.
  - La segunda manera es conectando los vértices cuya relación es mutua. Esto quiere decir que dos vectores $i, j \in V$ van a estar conectados si i se encuentra dentro de los k vecinos más cercanos de j **y** j se encuentra dentro de los k vecinos más cercanos de i.

  En este caso los pesos si aportan información adicional. Los pesos de las aristas son iguales a las similaridades entre los vértices.
- **Fully Connected graph.** Conecta todos los vértices con similaridades positivas y les asigna los pesos $w_{i,j} = s_{i,j}$. Ahora, como dijimos antes, el grafo debe poder representar el entorno local de cada vértice, y es por eso que necesitamos definir una función de similaridad que nos permita modelarlo.

  La función de similaridad Gaussiana es una de las funciones de similaridad más utilizadas:

$$s(x_i, x_j) = \exp\left(\frac{-||x_i - x_j||^2}{2\sigma^2}\right)$$

1.2. Grafos Laplacianos

Suponemos que tenemos un grafo no dirigido, G, con matriz de adyacencia W.

1.2.1. Grafos no-normalizados

Definimos al **grafo Laplaciano no-normalizado** de la siguiente manera:

$$L = D - W = \begin{pmatrix} d_{1,1} - w_{1,1} & \dots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \dots & d_{n,n} - w_{n,n} \end{pmatrix}$$

[1] Esto quiere decir que si el vértice j se encuentra dentro de los k vecinos más cercanos del vértice i, eso no quiere decir que el vértice i se encuentra dentro de los k vecinos más cercanos de j.

Notar que L no depende de los elementos en diag(W). Esto quiere decir que cualquier matriz de adyacencia que tenga los mismos elementos que W por fuera de la diagonal nos devuelve el mismo grafo L.

Estos grafos tienen tres propiedades importantes:

- Para todo vector $u \in \mathbb{R}^n$ se cumple que

$$u'Lu = \frac{1}{2}\sum_{i,j=1}^{n} w_{i,j}\,(f_i - f_j)^2$$

- L es simétrica y definida semi-positiva. Esto implica que el autovalor mas pequeño es $\lambda = 0$, el cual esta asociado al autovector de unos, $\mathbb{1}$.
- L tiene n autovalores no negativos, $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

Proposición 1. *Sea G un grafo no dirigido con pesos no negativos. Entonces, la multiplicidad k del autovalor $\lambda = 0$ del grafo Laplaciano no-normalizado (L) es equivalente al número de componentes conexas $A_1, \ldots, A_k$. Además, los autovectores asociados a $\lambda = 0$ son los vectores indicadores $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ de esas componentes.*

La Proposición 1 nos dice que la cantidad de clusters en nuestro conjunto de datos X es equivalente a la multiplicidad del autovalor $\lambda = 0$.

Para entender la idea detrás, supongamos primero que $k = 1$. Si u es un autovector con autovalor $\lambda = 0$, entonces se cumple que:

$$u'Lu = \sum_{i,j=1}^{n} w_{i,j}\,(u_i - u_j)^2 = 0$$

Si dos vértices están conectados, entonces $w_{i,j} > 0$. Por lo tanto, para que se cumpla la ecuación anterior, se tiene que cumplir que $(u_i - u_j)^2 = 0$, es decir, $u_i = u_j$. Ahora, como todo par de vértices dentro de una componente conexa está conectado (i.e., $w_{i,j} > 0$), necesitamos que $u_i = u_j = u$ para todo par de vértices i, j.

Esto último quiere decir que para un grafo con una única componente conexa solo tenemos un único vector indicador $\mathbb{1}$ como autovector cuyo autovalor es $\lambda = 0$. Es trivial que este autovector es el vector indicador de la componente conexa.

Ahora supongamos que tenemos $k > 1$ componentes conexas. Sin perder generalidad podemos asumir que los vértices se encuentran ordenados de acuerdo a la componente conexa a la que pertenecen[2]. En este caso, la matriz L tiene una forma diagonal en bloque:

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}$$

[2]Por ejemplo, supongamos que $k = 2$ y cada componente conexa tiene n_1 y n_2 vértices asociados, respectivamente. Luego, los vértices $\{v_1, v_2, \ldots, v_{n_1}\}$ pertenecen a la primera componente conexa y los vértices $\{v_{n_1+1}, \ldots, v_n\}$ pertenecen a la segunda componente conexa.

donde cada bloque L_i es un grafo Laplaciano correspondiente a la i-ésima componente conexa. L es una matriz de $n \times k$, donde $n = \sum_{i=1}^{k} n_i$, con n_i el tamaño de L_i.

Se cumple que los autovectores de L son los autovectores de L_i, completando con ceros en las posiciones de los demás bloques. Por ejemplo, el primer autovector de L, asociado con la primera componente conexa, tiene 1 en los primero n_1 elementos y cero en los restantes.

Por lo tanto, la matriz L tiene tanto autovalores $\lambda = 0$ como componentes conexas, y sus correspondientes autovectores son los vectores indicadores $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$.

2. Algoritmo

El algoritmo se puede resumir en los siguientes pasos:

1. Dado nuestro conjunto de datos, $\{x_1, \ldots, x_n\}$, que en principio pueden representar cualquier objeto, calculamos la matriz (grafo) de similaridad, $S = (s_{i,j})_{i,j=1,\ldots,n}$, utilizando alguno de los métodos mencionados en la Sección 1.1.2. También calculamos la matriz de adyacencia, W.
2. Computamos la matriz (grafo) Laplaciano no-normalizado, $L = D - W$.
3. Computamos los primeros k autovectores $\{u_1, \ldots, u_k\}$ de la matriz L[3] y llamamos $U \in \mathbb{R}^{n \times k}$ a la matriz que contiene a los autovectores en sus columnas.
4. Para todo $i = \{1, \ldots, n\}$ definimos $y_i \in \mathbb{R}^k$ como el vector que representa a la i-ésima fila de U.
5. Finalmente, ajustamos cualquier método de clustering sobre los puntos $y_i \in \mathbb{R}^k$ para obtener la partición $A_1, \ldots, A_k$.

Lo poderoso de este algoritmo es el cambio de representación de las observaciones x_i a los puntos $y_i \in \mathbb{R}^k$. Este cambio de representación mejora las propiedades de los clusters en nuestro conjunto de datos, lo que hace que sea trivial detectarlos.

[3]Por "primeros k autovectores" nos referimos a los autovectores asociados a los k autovalores más pequeños.