Métodos Numéricos II 2024

Primer Proyecto

06.septiembre.2024

Segmentación Espectral (Spectral Clustering).

En este proyecto vamos a implementar un método para binarizar (segmentar en dos componentes) una imagen RGB.

A partir de una imagen a colores (por ejemplo, en formato RGB), en general la segmentación consiste en clasificar cada píxel de la imagen en k grupos, atendiendo a características comunes como color, textura y posición espacial. En particular, la segmentación binaria consiste en clasificar los píxeles en dos grupos: 0 = fondo (*background*), y 1 = objeto de interés (*foreground*).
Una técnica para segmentación binaria consiste en la segmentación espectral.

Dada la imagen I de $h \times w$ píxeles (h = número de filas o altura de la imagen, w = número de columnas o ancho de la imagen), el método de segmentación espectral construye un grafo G con hw vértices, uno por cada píxel de la imagen.

Cada píxel es conectado con sus cuatro vecinos (norte, sur, este y oeste), y las aristas de conexión se ponderan de acuerdo a la similitud de los píxeles vecinos que se unen, mediante alguna métrica de afinidad. Por ejemplo, una de las métricas de afinidad más comunes se construye mediante un kernel gaussiano

$$similarity(\mathbf{p}_i, \mathbf{p}_j) = \exp\Big(-\frac{1}{2\sigma^2}(\alpha||color(i) - color(j)|| + \beta||position(i) - position(j)||)\Big).$$

Aquí, $color(i)$ y $color(j)$ corresponden al vector de colores RGB de los píxeles $\mathbf{p}_i$ y $\mathbf{p}_j$, respectivamente. Similarmente, $position(i)$ y $position(j)$ son las coordenadas (x, y) de los píxeles $\mathbf{p}_i$ y $\mathbf{p}_j$, en la matriz $h \times w$ que almacena la imagen I.

Así, $||color(i) - color(j)||$ mide la distancia de color entre los píxels i y j, mientras que $||position(i) - position(j)||$ mide la distancia espacial entre estos píxeles. $\alpha > 0$ y $\beta > 0$ son coeficientes o parámetros que ponderan la diferencia de color y de distancia espacial, y $\sigma > 0$ es un parámetro de escala dependiendo de la imagen.

1. El primer paso de la segmentación espectral es construir una matriz de afinidad W, que representa al grafo de conexiones G, pesado mediante la métrica de afinidad anterior. Esta matriz $W \in \mathbb{R}^{hw \times hw}$ es simétrica, y usualmente es de dimensiones muy grandes, con pocas conexiones. Así que W se trabaja como una matriz rala.

2. Construida la matriz de afinidad $W = (w_{ij})$, se construye la matriz Laplaciana del grafo G. Para ello, en cada fila de la matriz W se calcula
$$d_i = \sum_{j=1}^{hw} w_{ij}.$$
Esto es, d_i es la suma de afinidades en la fila i (d_i se llama usualmente el **grado** del vértice i). Luego, se construye la matriz diagonal
$$D = diag(d_1, d_2, \ldots, d_{hw}).$$
El Laplaciano del grafo G se define como la matriz
$$L = W - D.$$
Sin embargo, en lugar de L, se utiliza el llamado *Laplaciano normalizado* de G, definido como
$$\mathcal{L} = D^{-1/2} L D^{-1/2} = D^{-1/2}(W - D)D^{-1/2}.$$

3. Con el Laplaciano normalizado $\mathcal{L}$, calculamos el autovector asociado al segundo menor autovalor de $\mathcal{L}$ (el menor autovalor siempre es 0). Este autovalor $\mathbf{v}_{n-1}$ es llamado el **vector de Fiedler**.

4. Finalmente, los signos del vector de Fiedler $\mathbf{v}$ producen la segmentación binaria requerida. Esto es, si la i-ésima entrada v_i del vector de Fiedler satisface $v_i > 0$, el píxel i de la imagen se asigna al grupo 1, mientras que si $v_i < 0$, el píxel i se asigna al grupo 0. Esto se hace para cada uno de los píxeles de la imagen. Al final de este paso, se obtiene una segmentación *background-foreground* de la imagen I.

Su trabajo consiste en implementar un método que haga una segmentación espectral a partir de la matriz de afinidad W.

- A partir de la imagen de entrada I de tamaño $h \times w$, construir el grafo de pixeles G de tamaño $hw \times hw$. Este debe ser representado por una matriz rala con 4 diagonales.
- Usando una métrica de similaridad con kernel gaussiano, construir una matriz de afinidad W, también en formato sparse.
- Calcular la laplaciana L y la laplaciana normalizada $\mathcal{L}$ en formato sparse.
- Hallar el segundo menor autovalor de $\mathcal{L}$ y su vector de Fiedler $\mathbf{v}$.
- Calcular los signos de las entradas de $\mathbf{v}$ y construir una máscara binaria en función de estos signos.
- Mostra la imagen original, y su segmentación binaria respectiva.

Repetir esto para las diferentes imágenes.

Material de consulta:
`http://www.tml.cs.uni-tuebingen.de/team/luxburg/publications/Luxburg07_tutorial.pdf`
`https://ranger.uta.edu/~chqding/Spectral/`