SEGMENTACIÓN DE IMÁGENES A COLOR MEDIANTE TÉCNICAS DE AGRUPAMIENTO DE DATOS EMPLEANDO LOS ALGORITMOS K-MEANS Y C-MEANS

Eloisa Guevara Gómez Omar *Alejandro Sánchez Tello* *Yesenia Eleonor González Navarro, Dra.*

Instituto Politécnico Nacional

UPIITA

guevara.elois@gmail.com osanllet@gmail.com

Abstract

This article describes the K -Means and C -Means (also known as Fuzzy C-Means) algorithms, which are two of the main data clustering algorithms. An application of these techniques in image segmentation is presented.

Key Words: color image segmentation, cluster, clustering, K-Means, C-Means, Fuzzy C-Means.

Introducción

La segmentación de imágenes es el proceso de seleccionar y agrupar todos aquellos pixeles (datos) que cuentan con características visuales y númericas similares entre sí. Al realizar la agrupación de los pixeles se generan cúmulos (*clusters*) de datos de una imagen, cada cúmulo está definido por un centro, el cuál representa un valor (en el caso de las imágenes, un color) con características similares a los datos que pertenecen a dicho agrupamiento.

El agrupamiento de datos (c*lustering*) es una técnica común para la clasificación de datos que permite tener un mejor manejo de ellos; esta técnica se utiliza en diversos campos como la identificación de estructuras en modelos difusos [1], compresión de datos y la segmentación de imágenes, donde la distribución de la información puede ser de cualquier tamaño y forma.

Algortimo K-Means

El algoritmo ***K-Means*** es uno de los métodos de clustering iterativos más populares. Se considera un conjunto de *n* datos a clasificar dentro del universo *X*:

Cada punto será asignado a uno y sólo un clúster, de tal forma que también se les denomina particiones. La función característica es definida como:

$$\chi_{A_i}(x_k) = \begin{Bmatrix} 1 & x_k \in A_i \\ 0 & x_k \notin A_i \end{Bmatrix} \ldots (2)$$

Una asignación de pertenencia del punto j en el clúster *i* se define como:

$$\chi_{ij} = (i = 1,2,\ldots,c;\ j = 1,2,\ldots,n), \ldots (3)$$

donde c es el número de cúmulos. Un espacio para una partición para *X* se define como la siguiente matriz:

$$M_c = \left\{ U | \chi_{ij} \in \{0,1\}, \sum_{i=1}^{c} \chi_{ij} = 1, 0 < \sum_{k=1}^{n} \chi_{ik} < n \right\}, \ldots (4)$$

donde U es una matriz de *c* renglones y *n* columnas.

El algoritmo propuesto para las particiones es una aproximación a partir de la suma de los errores cuadráticos empleando una norma euclidiana para caracterizar la distancia entre dos puntos.

La función costo se define entonces como:

$$J(U,v) = \sum_{k=1}^{n} \sum_{i=1}^{c} \chi_{ik} (d_{ik})^2, \ldots (5)$$

donde v es un vector que contiene los centros de cúmulo y d_{ik} es una distancia euclidiana medida en un espacio de *m* dimensiones entre el punto k y el centro v_i del cúmulo i, descrita de la forma:

$$d_{ik} = d(x_k - v_i) = \|x_k - v_i\| = \left[\sum_{j=1}^{m} (x_{ij} - v_{ij})^2 \right]^{1/2} \ldots (6)$$

Los *m* elementos del vector *vi* v_i se calculan a partir de:

$$v_{ij} = \frac{\sum_{k=1}^{n} \chi_{ik} \cdot \chi_{kj}}{\sum_{k=1}^{n} \chi_{ik}} \ldots (7)$$

Para encontrar la partición óptima U^* que produce el valor mínimo de la función costo *J* es necesario emplear el siguiente método iterativo:

1. Fijar *c* tal que $2 \leq c < n$ e inicializar la matriz *U*.
2. Calcular los vectores de cada centro para los c conjuntos: $\{v_i^{(r)}$ con $U^{(r)}\}$, r=1,2,... iteraciones.
3. Actualizar $U^{(r)}$ y calcular las funciones características actuales para toda i, k:

$$\chi_{ik}{}^{r+1} = \begin{Bmatrix} 1 & d_{ik}{}^{(r)} = \min\{d_{jk}{}^{(r)}\} para\ toda\ j \in c \\ 0 & otros\ casos \end{Bmatrix} \ldots (8)$$

4. Si $\|U^{(r+1)} - U^{(r)}\| \leq \varepsilon$, donde ε es el nivel de tolerancia, detener, si no establecer *r = r + 1* y regresar al paso 2.

Algoritmo C-Means o Fuzzy C-Means.

El algoritmo ***Fuzzy C-Means*** es una extensión del algoritmo K-Means. Mientras K-Means encuentra distribuciones para las que un punto pertenece a un solo cúmulo, Fuzzy C-Means es un método que encuentra centros de cúmulos donde un punto puede pertenecer a más de uno con un cierto grado de pertenencia.

En el caso de los cúmulos difusos, se define una familia de conjuntos difusos *{Ai, i = 1,2,...,c}* en un universo de puntos X.

Es posible asignar una pertenencia a los distintos datos en cada conjunto difuso, de tal forma que el punto k tiene la siguiente pertenencia a la clase i:

$$\mu_{ik} = \mu_{\widetilde{A_i}}(x_k) \in [0,1] \ldots . (9)$$

La matriz de particiones difusas se define como:

$$M_{fc} = \left\{ \widetilde{U} | \mu_{ij} \in [0,1];\ \sum_{i=1}^{c} \mu_{ij} = 1;\ 0 < \sum_{k=1}^{n} \mu_{ik} < n \right\}, i = 1,2, \ldots, c, k = 1,2, \ldots, n. \ldots (10)$$

Para este algoritmo se define una función objetivo J_m con una matriz de partición $\widetilde{U}$ para agrupar n datos en c clases:

$$J_m(\widetilde{U}, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^{m'} (d_{ik})^2, \ldots (11)$$

donde:

$$d_{ik} = d(x_k - v_i) = \left[\sum_{j=1}^{m} (x_{ij} - v_{ij})^2 \right]^{1/2} . \ldots (12)$$

El parámetro m' controla la cantidad de difusificación en el proceso de clasificación, y aunque puede tener valores entre $[1, \infty)$, normalmente se emplean valores de 1 o 2. Los centros de cada cúmulo se calculan con la expresión:

$$v_{ij} = \frac{\sum_{k=1}^{n} {\mu_{ik}}^{m'} . x_{kj}}{\sum_{k=1}^{n} {\mu_{ik}}^{m'}} . \ldots (13)$$

A continuación se describe el algoritmo:

1. Fijar c tal que $2 \leq c < n$, seleccionar m e inicializar la matriz $\widetilde{U}$.
2. Calcular los centros $\left\{ v_i^{(r)} \right\}$ para los c conjuntos, r=1,2,... iteraciones.
3. Actualizar la matriz de partición $\widetilde{U}^{(r)}$ para la iteración r.

$$\mu_{ik}{}^{(r+1)} = \begin{Bmatrix} 1 & \left[\sum_{j=1}^{c} \left(\frac{d_{ik}{}^{(r)}}{d_{jk}{}^{(r)}} \right)^{2/(m'-1)} \right]^{-1}, para\ I_k = \phi \\ 0 & para\ todas\ las\ clases\ i\ donde\ i \in \tilde{I}_k \end{Bmatrix} . \ldots (14)$$

Con $I_k = \{i | 2 \leq c < n;\ d_{ik}{}^{(r)} = 0\}$; $\tilde{I}_k = \{1,2,\dots,c\} - I_k \sum_{i \in I_k} \mu_{ik}{}^{(r+1)} = 1$.

5. Si $\left\| \tilde{U}^{(r+1)} - \tilde{U}^{(r)} \right\| \leq \varepsilon$, donde ε es el nivel de tolerancia, detener, si no establecer $r = r + 1$ y regresar al paso 2.

Desarrollo

Los algoritmos anteriormente descritos fueron implementados en programas de MATLAB®. Como prueba, se empleó la imagen de la Figura 1 con extensión .jpeg en formato de color RGB y resolución 243 x 320 pixeles. Al final de este artículo se encuentra un vínculo para descargarlos.

Figura 1. Imagen en formato de color RGB a segmentar.

La Figura 2 muestra la dispersión de los pixeles en el espacio de color RGB. Cada eje representa una componente de color. También se muestra la posición de los centros de cúmulos obtenidos por los algortimos K-Means y C-Means.

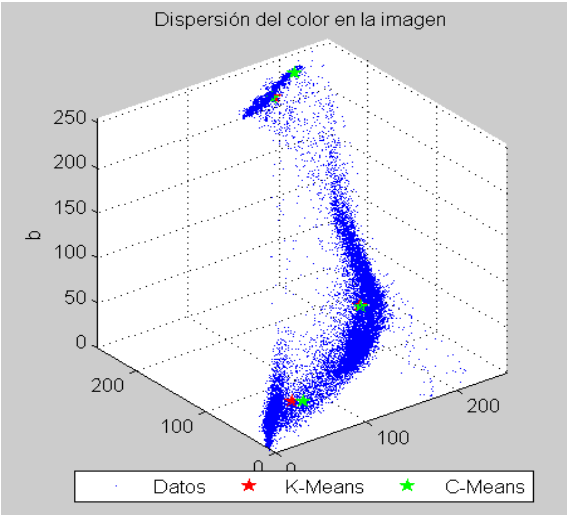

Figura 2. Dispersión de los pixeles en la imagen y centros de cúmulos obtenidos por los algoritmos K-Means y C-Means.

La Figura 3 muestra la imagen segmentada mediante los algoritmos K-Means y C-Means. Para la obtención de las imagenes segmentadas (reconstrucción de imagen), se realizó un barrido de los pixeles de la imagen original y se sacó la distancia euclidiana a cada centro de cúmulo (3 componentes). Los valores para cada pixel se sustituyeron por los valores del centro de cúmulo más cercano. Otra opción es asignar los valores RGB a cada pixel reconstruido de acuerdo a la matriz *U* de partición final.

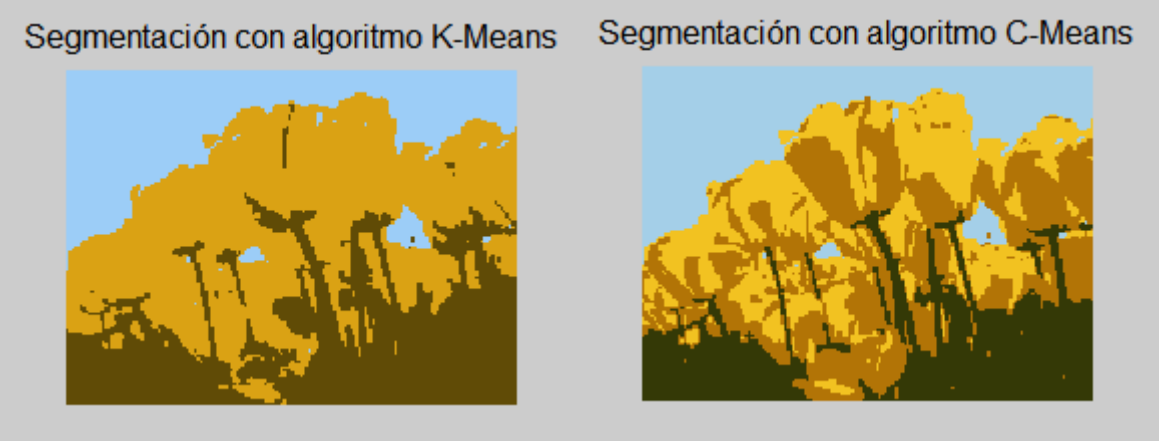

Figura 3. Resultados de segmentación con los métodos K-Means y C_Means.

De acuerdo a los valores numéricos establecidos para las variables, K-Means encontró los centros de 3 cúmulos y C-Means encontró 4. Cada cúmulo representa un color, de esta forma la imagen queda dividida en determinado número de partes y es posible clasificar sus componentes. A partir de un análisis visual de las imágenes segmentadas que se obtuvieron C-Means arroja mayor definición, ya que al encontrar un cúmulo más que en K-Means la imagen segmentada resultante arroja mayor detalle en el área de las flores, pero si lo que se requiere es separar objetos o regiones, C-Means separó fondo, tallos y flores; por lo que depende de la aplicación la selección del algoritmo a utilizar.

Conclusiones

Los algoritmos K-Means y C-Means son una herramienta útil para encontrar los centros de cúmulos de un determinado grupo de datos. La elección del algoritmo a utilizar dependerá de la aplicación. El desempeño de ambos algoritmos depende de la posición inicial de los centros de cada cúmulo para que el resultado converja a la solución óptima y se requiere conocer de antemano el número de éstos. También es necesario elegir un valor adecuado para la condición de paro ε y en el caso de C-Means, el valor de *m'*. Asimismo, es conveniente obtener valores aproximados a estos centros mediante otros algoritmos como los descritos en el artículo "**SEGMENTACIÓN DE IMÁGENES A COLOR EMPLEANDO LOS MÉTODOS DE LA MONTAÑA Y SUSTRACTIVO PARA LA ACELERACIÓN DEL PROCESAMIENTO DE LOS ALGORITMOS K-MEANS Y C-MEANS" [4].**

Referencias

1. Ponce Cruz, Pedro (2010). *Inteligencia Artificial con Aplicaciones a la Ingeniería* (1ª Edición). México: Alfaomega. ISBN: 978-607-7854-83-8.
2. Jang J. S., Sun C. T., Mitzuani E. (1997). *Neuro Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. USA: Prentice Hall Inc. ISBN: 978-0132610667.
3. Hung, MC, Yang, D. L. (2001). *An Efficient Fuzzy C -Means Clustering Algorithm.* USA: Proceedings IEEE International Conference on Data Mining. Pag. 225-232. ISBN: 0-7695-1119-8.

4. Guevara Gómez E., Sánchez Tello O., González Navarro Y. (2015). *Segmentación de Imágenes a Color Empleando los Métodos de la Montaña y Sustractivo para la Aceleración del Procesamiento de los Algoritmos K-Means y C-Means*. México: Boletín UPIITA, No. 51. ISSN: 2007-6150.

Descarga de programas en Matlab®

http://www.upiita.ipn.mx/index.php/academias/75-profesores/sistemas/241-material-didactico-m-en-c-yesenia-eleonor-gonzalez-navarro