SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de León

División de Estudios de Posgrado e Investigación

Maestría en Ciencias de la Computación

MATERIA

Reconocimiento de Patrones

PRESENTA

Ing. José Alejandro Cornejo Acosta

CATEDRÁTICO

Dr. Alfonso Rojas Domínguez

León, Guanajuato. **1 de Diciembre de 2016**

Introducción

[CITATION "ALGORITMOS DE APRENDIZAJE"]

Dentro de los tipos de aprendizaje, está el aprendizaje supervisado y el aprendizaje no supervisado, el aprendizaje supervisado es cuando tenemos las etiquetas de nuestro conjunto de entrenamiento, cuando no tenemos las etiquetas, tenemos que utilizar algún método de clustering (agrupamiento) para poder dividir nuestro conjunto de datos en *N* clases, existen varios algoritmos para realizar esto, uno de los más comunes y el que se utilizó para este trabajo es el algoritmo *k-means,* este algoritmos fue creado por MacQueen en 1967, es el más conocido por que su implementación es simple y sus resultados son eficaces.

Una posterización de imagen, consiste en reducir la profundidad de bits de una imagen tal que tiene un impacto visual, normalmente se utiliza para quitar las sombras y detalles de las imágenes, muchos software como *photoshop* ya tienen implementada esta funcionalidad. Una desventaja de la posterización, es se puede perder mucha calidad de la imagen durante el procesamiento, ya que la cantidad de colores presentes disminuye.

Objetivos

- Entender el concepto de clusterización (agrupamiento) y posterización de imágenes.
- Implementar el algoritmo K-means.
- Utilizar las funciones *reshape* para la conversión de imágenes a matrices y viceversa del lenguaje de programación Matlab.

Desarrollo

Pasos para el algoritmo de K-means

1. Seleccionar *k* elementos aleatorios de nuestro conjunto que serán nuestros clusters y centroides iniciales.
2. Medir la distancia de cada elemento del conjunto de datos a los *k* clusters, y etiquetar este elemento con la clase del cluster más cercano.
3. Actualizar los valores de los clusters, para actualizar cada cluster, deben promediarse los elementos que pertenecen a ese cluster.
4. Volver a mediar la distancia de cada elemento del conjunto de datos a los *k* nuevos clusters y volver a etiquetar los elementos con la clase del cluster más cercano.
5. Verificar si hubo elementos dentro del conjunto que cambiaron la clase a la que pertenecían, si hubo cambio, repetir a partir del paso 3 hasta que dejen de haber cambios entre las etiquetas del los elementos del conjunto.

Una de las preguntas más frecuentes que hay al momento de hacer una implementación de este algoritmo, es cuál es el valor óptimo para *k*, existe algunos métodos para calcular que factible es un determinado valor de *k* para este algoritmo, el método que se utilizó en este trabajo es el más simple, el método de suma de cuadrados también conocido como método de codo, este método se define de la siguiente manera:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(\mu, x)$$

donde:

K = Es el total de clusters

C= Es un cluster

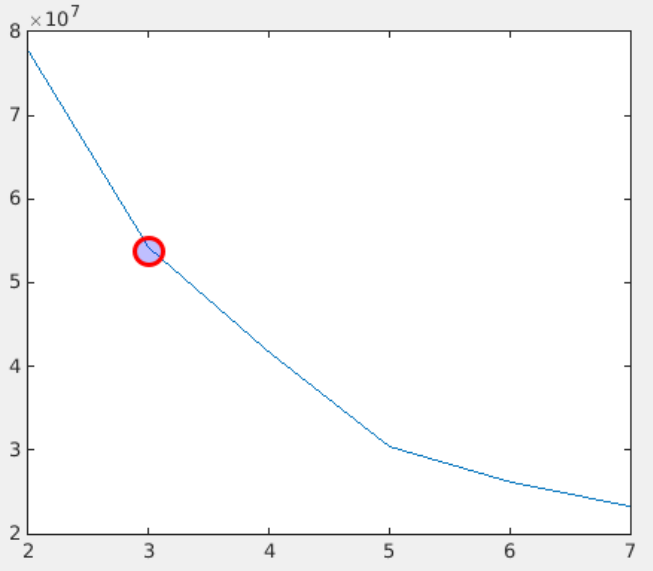
μ = Es el promedio del cluster C

x = Es una instancia del cluster C

$$dist^2 = \sum_{i=1}^{n} (x_{1i} - x_{2i})^2$$

NOTA: En el algoritmo de *k-means*, lo más común es utilizar la distancia *Euclidiana* como medida, (y es la que se utiliza en este trabajo) pero el algoritmo no se limita a eso, también pueden utilizarse otras medidas de distancia. Es por eso que en la función $dist^2$, es igual a la distancia Euclidiana elevada al cuadrado.

A este método se le conoce como método de codo por que si se grafica para un intervalo de valores *k,* se obtiene una gráfica similar a la siguiente:



En el eje X, se grafica los valores de *k*, y en el eje Y, son los valores de la suma de cuadrados que se obtuvieron para cada valor, el método nos dice que donde esté un el cambio de pendiente más resaltado, es donde se encuentra el valor *k* adecuado para la clusterización. Las ventajas de este método, es que es fácil de implementar y no tiene mucho costo computacional, la desventaja es que si la curva generada es muy suave, entonces se puede volver ambiguo este método.

Resultados

En este trabajo, se trabajó con una imagen en formato jpg de tamaño *600 x 468 pixeles*, que fue escalada a un 25% para que el tiempo de ejecución del algoritmo no fuera demasiado alto.

La imagen original es:

Se hicieron pruebas para diferentes valores de *k* y se repitieron los experimentos para obtener un promedio de los valores del método suma de cuadrados.

Para *k = 2,* la imagen procesada que se obtuvo es la siguiente:

Para *k = 3*

Para *k = 4*

Para *k = 5*

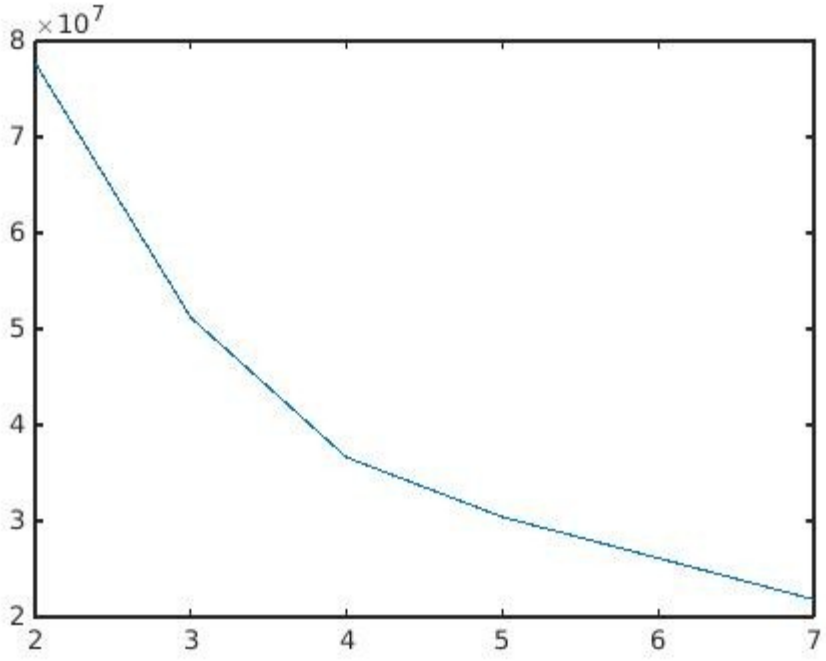
Para *k=6*

Para *k=7*

Se puede observar que mientras k sea mayor, la imagen es mas parecida a la original, esto nos podría decir, que al menos para este caso, mientras mayor sea el valor k, la clusterización es mejor, pero esto no necesariamente es verdad, también se hizo un análisis con las gráficas del método de suma de cuadrados.

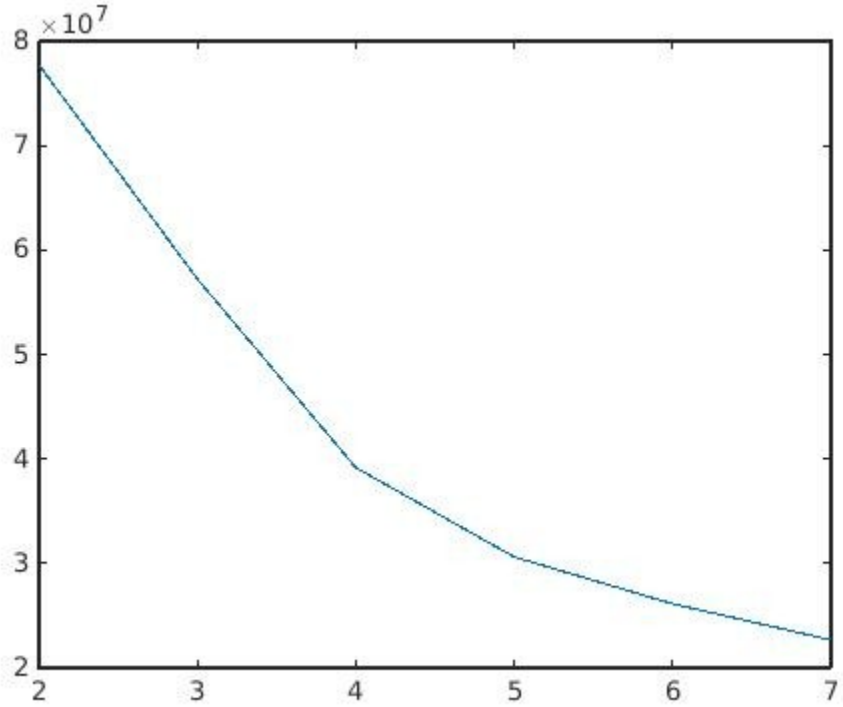
Como los clusters iniciales del algoritmo k-means se generan se forma aleatoria, puede que el resultado de suma de cuadrados varíe con cada experimento, enseguida se muestran unas gráficas sobre el método de suma de cuadrados para los distintos valores de *k*.

Para 1 iteración:



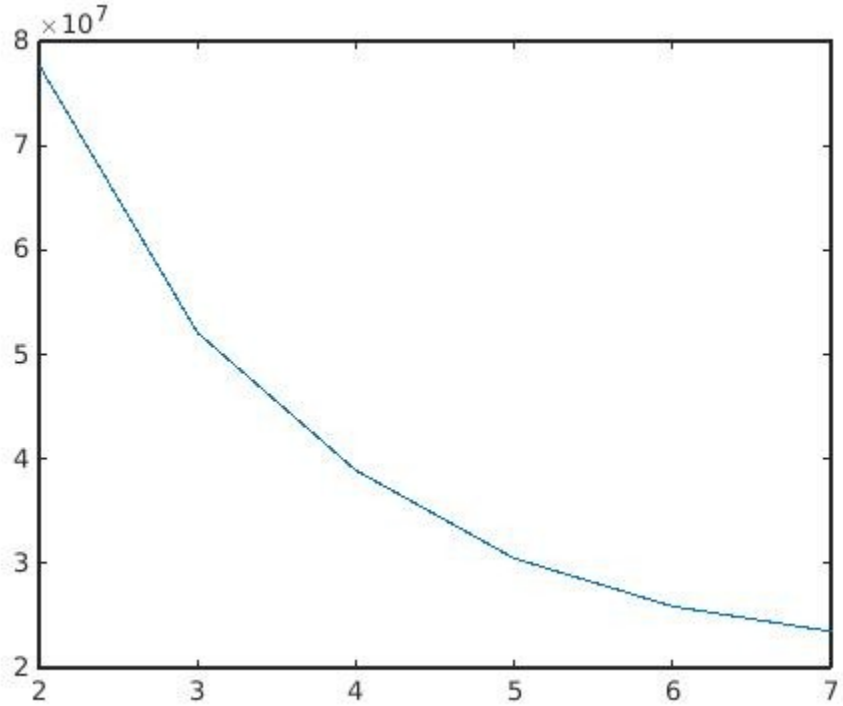
Con una iteración, con el método de suma de cuadrados podemos concluir que el valor óptimo de k es 3, ya que es donde se nota la curva.

Para 3 iteraciones promediadas:



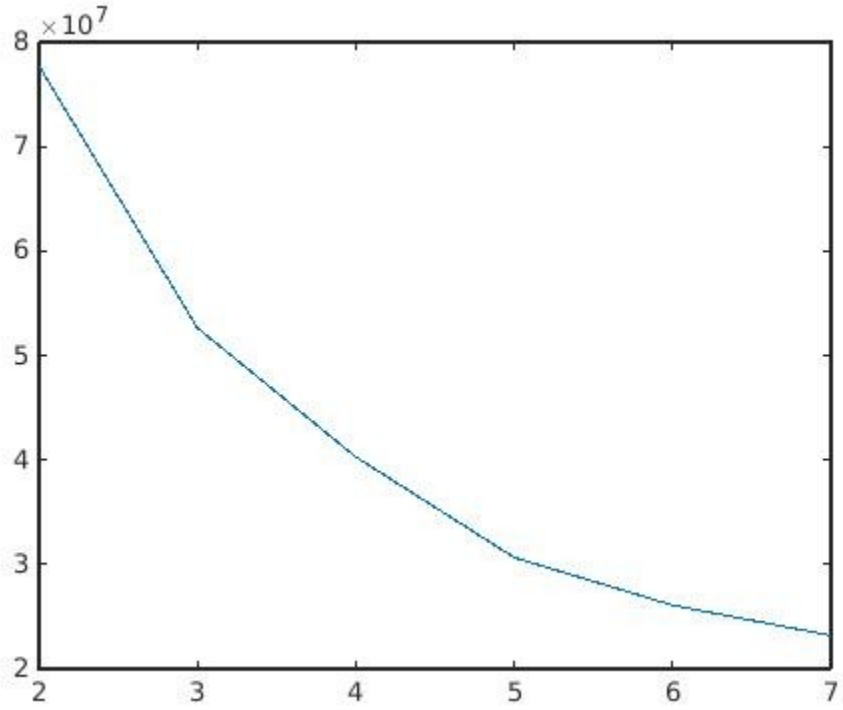
A diferencia del ejemplo anterior con una sola iteración, en este caso la gráfica nos indica que el valor óptimo de *k* es 4.

Para 10 iteraciones promediadas:



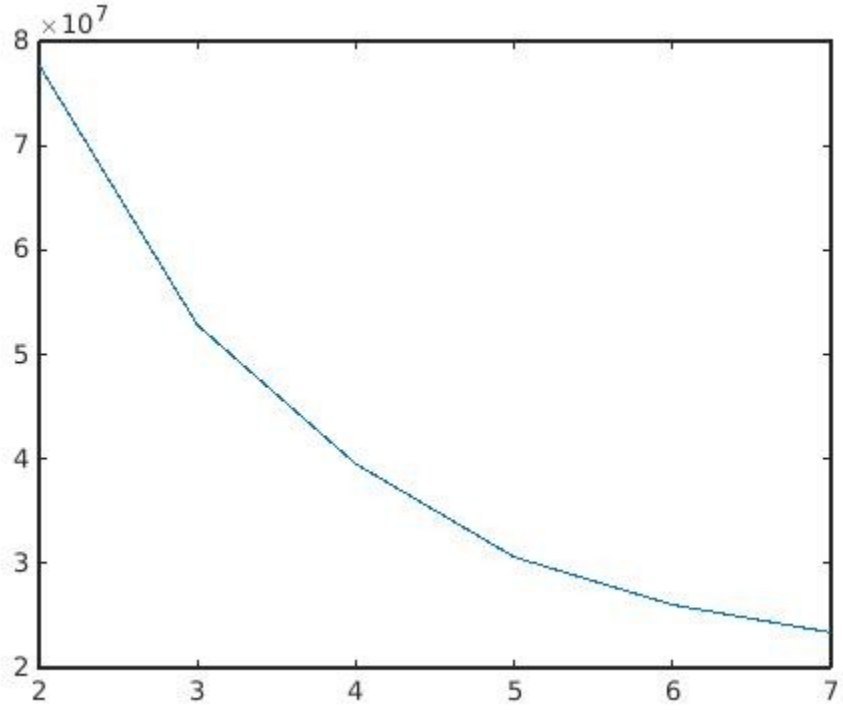
En este caso, podemos observar que nuevamente el valor óptimo de *k* es 3.

Para 25 iteraciones promediadas:



Se puede observar que la curva no cambia mucho a diferencia de los casos anteriores donde se realizaban menos iteraciones.

Para 50 iteraciones promediadas:



En las gráficas se puede notar que mientras mayor sea la cantidad de experimentos (iteraciones), la curva de suma de cuadrados es más precisa. En todos los casos, los resultados nos arrojan que el valor óptimo de *k* para este caso es 3 o 4, casi todos los experimentos nos muestran que es 3.

Otra cosa importante a mencionar, es que el tamaño de la imagen es muy importante, no solo porque influye en el tiempo de ejecución de los algoritmos, si no también porque si la imagen es más grande obviamente la posterización es de mejor calidad, esto se vio en un experimento que se realizó con una imagen no tan reducida y con *k = 7*.

Conclusiones

En este trabajo se aplicó el clustering (agrupamiento) la posterización de imágenes, pero también se puede aplicar en diferentes áreas de Ciencias de la Computación, en minería de datos también es común trabajar con este tipo de algoritmos. En el área optimización combinatoria también se pueden implementar algoritmos de agrupamiento, un ejemplo es el problema del viajero colaborativo, en donde se tiene un conjunto de ciudades que se tienen que visitar, entonces entre dos agentes o más se reparten las ciudades que visitará cada uno pero se las tienen que repartir de manera óptima, para esto las ciudades se tienen que clusterizar. Otra aplicación puede ser en el estudio de los terremotos, la reagrupación de los epicentros de los sismos observados, permite determinar las zonas de riesgos, y poder ayudar a evitar catástrofes o a mitigar el daño ocurrido. Durante el desarrollo de este trabajo, también se pudo observar que el costo computacional de los algoritmos de clustering en imágenes es muy ato, ya que en los casos donde k = 7 el tiempo de ejecución era de varios segundos, y al repetir este experimento 20, 30 o 50 veces, es un tiempo muy significante, por lo que sería interesante analizar como funcionan algunos software como photoshop que pueden posterizar imágenes de muy alta calidad en tiempos muy cortos.

Bibliografía

ALGORITMOS DE APRENDIZAJE: Cristina García Cambronero, Irene Gómez Moreno,
ALGORITMOS DE APRENDIZAJE: KNN & KMEANS,