Laboratorio 03: kMeans: algoritmo de clustering

Por:

Ian Gabriel Cañas Fernández, 1092228

.

Profesor: Juan S. Pérez R.,

Asignatura: INL367L, Secc 01

# Resumen:

El presente laboratorio estará apreciando el método por agrupamiento, que busca la partición por grupos de una cantidad dada de observaciones, cada una perteneciendo a un valor medio más cercano. Es decir, se procurará encontrar patrones e identificarlos en un dataset. Además, se procurará aplicar este método, llamado k-means, para realizar una segmentación por colores de una imagen.

# Ejercicios previos:

## P3.1. Algoritmo k-Means.

Para la búsqueda de sentido a un conjunto de datos sin etiquetas pero con distinción aparente, se reconoce el algoritmo k-Means, que procura la agrupación de los valores observados a una cantidad K de centroides que conforman buenas representaciones de datos. El funcionamiento de este algoritmo se ve plasmado por Alpaydin en la Ilustración 1:

Une image contenant texte

Description générée automatiquement

Ilustración 1. Psudocódigo de K-Means.

Este algoritmo da a entender que se ubican varios centros en posiciones al azar en el dataset y a partir de ello se va reajustando su posición en base a los grupos de datos cercanos a este. Dándole importancia a ciertas mediciones para cada centroide.

## P3.2. Empleo de K-Means para realizar una segmentación por colores de una imagen.

Para esta sección, busca aplicar el algorimo implementado en la sección anterior en la segmentación por colores de una imagen sustituyendo los píxeles de la imagen original por los centroides de los racimos (clústers) encontrados a los que pertenecen.

Además, se solicita comparar los resultados obtenidos mediante este método y mediante el uso de la función modefilt a un tamaño de filtro de 9x9, y así comparar los resultados que este obtenga.

# E3.1 Implementación en Matlab de K-Means.

En la **Erreur ! Source du renvoi introuvable.** se puede observar el algoritmo que genera la base de datos de las “observaciones” como creando una gráfica con grupos seccionados por categorías. En la Ilustración 3 se aprecia el algoritmo encargado de crear 5 centroides al azar a partir de cuyas coordenadas iniciará le reubicación de sus posiciones. En la Ilustración 4 se presenta el proceso utilizado para graficar los datos por colores según las distinciones que se tuvieron en

Une image contenant texte

Description générée automatiquement

Ilustración 2. Adaptación k-NN a varios tipos de distancias.

Une image contenant texte

Description générée automatiquement

Ilustración 3. Seteo y reagrupación de centroides.

Une image contenant texte

Description générée automatiquement

Ilustración 4. Graficado de datos.

# R3.1 Resultados:

En la Ilustración 5se puede apreciar el dataset generado para trabajar con este método y en la Ilustración 6 se aprecia en grandes rasgos varios resultados obtenidos volviendo a correr el mismo código para varios valores iniciales de centroides, todos generados al azar.

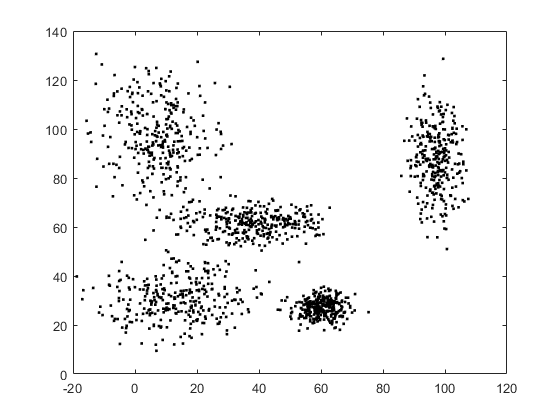


Ilustración 5. Dataset generado.

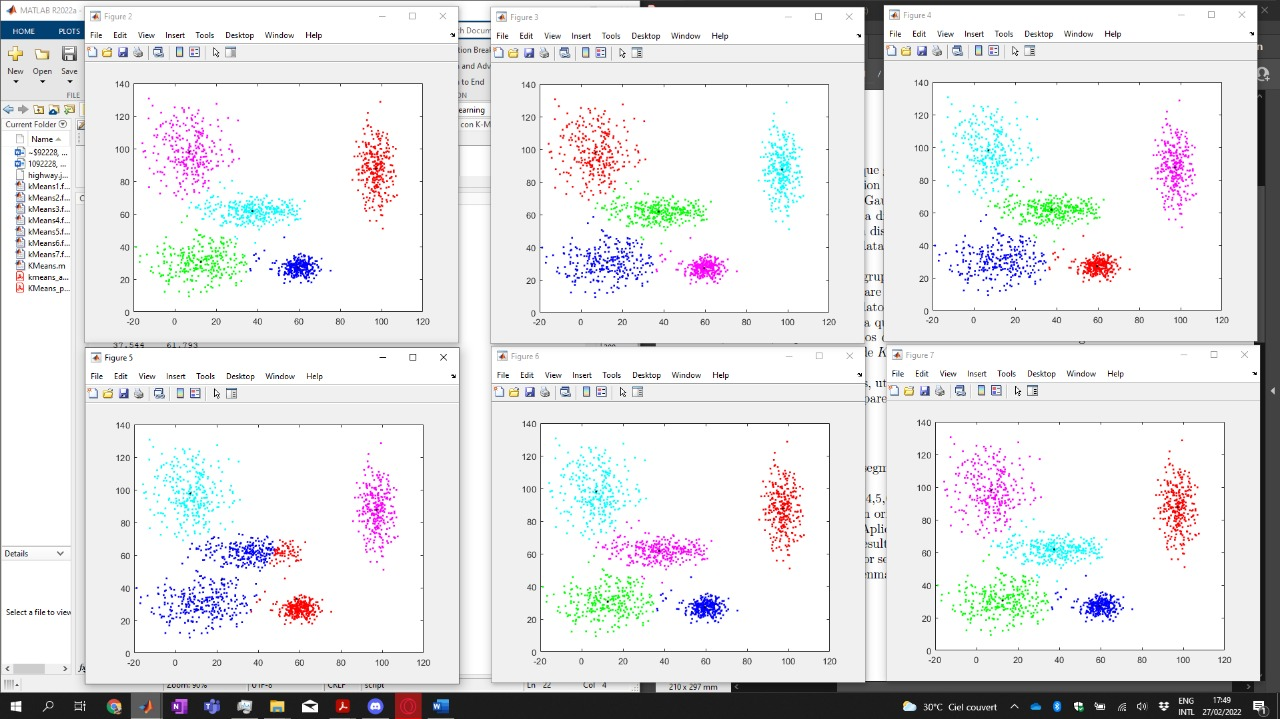


Ilustración 6. Implementación de K-Means para diferentes valores de centroides.

En la ilustración vista más atrás se puede apreciar que los valores de los centroides tienden a ubicarse en valores parecidos según sus posiciones iniciales, estas coordenadas y la cantidad de iteraciones que hayan tomado se presentan en Ilustración 7:

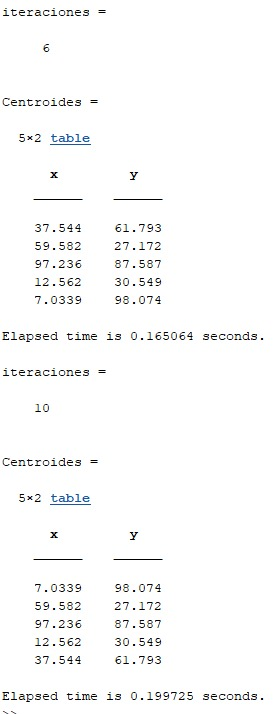
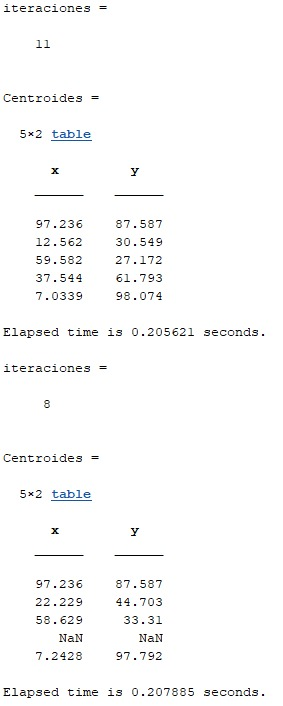
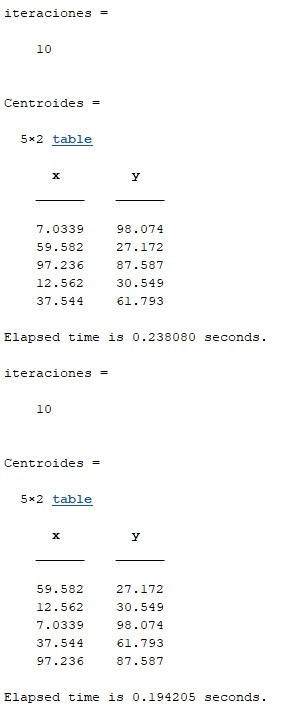


Ilustración 7. Centroides y cantidad de iteraciones para K-Means

# A.3.1 Análisis:

Se ha podido observar que los centroides en casi todas las iteraciones se presentan prácticamente en los mismos lugares, a excepción de ciertos casos en que un centroide se pierda por no ser el más cercano a ningún dato, pero al ser un caso poco probable podemos tener garantía o suficiente seguridad de que el algoritmo es suficientemente útil para reconocer la presencia de patrones y clasificarlos como tal. Se observa que este método tiene cierto comportamiento inverso a los kNN, pues este último crea patrones a partir de etiquetas y k-Means crea una etiqueta a partir de patrones.

# E3.2 Implementación en Matlab de K-Means.

En la Ilustración 8, Ilustración 9 y Ilustración 10 se puede observar el algoritmo implementado para el filtrado de la imagen, retomando el funcionamiento del algoritmo implementado en kMeans, siendo adaptado a tres dimensiones, generalizándose así a un espacio n-dimensional con k-clústers. Nótese que los datos han sido cambiados de uint8 a int32; 32 porque se llega a trabajar con valores mayores a 16 bits y con signos por los valores negativos que surgen en los cálculos en la línea 37.

Une image contenant texte

Description générée automatiquement

Ilustración 8. Preparación de datos.

Une image contenant texte

Description générée automatiquement

Ilustración 9. Reasignación de color según centroides.

Une image contenant texte

Description générée automatiquement

Ilustración 10. Muestreo de resultado.

# R3.2 Resultados:

En la Ilustración 11 se observa la imagen original; en la Ilustración 12 se observa el filtro de moda aplicado y en la Ilustración 13 a la Ilustración 17 se presentan los resultados aplicando kMeans para los valores de k = 3, 4, 5, 6.



Ilustración 11. Imagen original.



Ilustración 12. Filtro de moda aplicado a imagen.



Ilustración 13. Filtrado con .



Ilustración 14. Filtrado con .

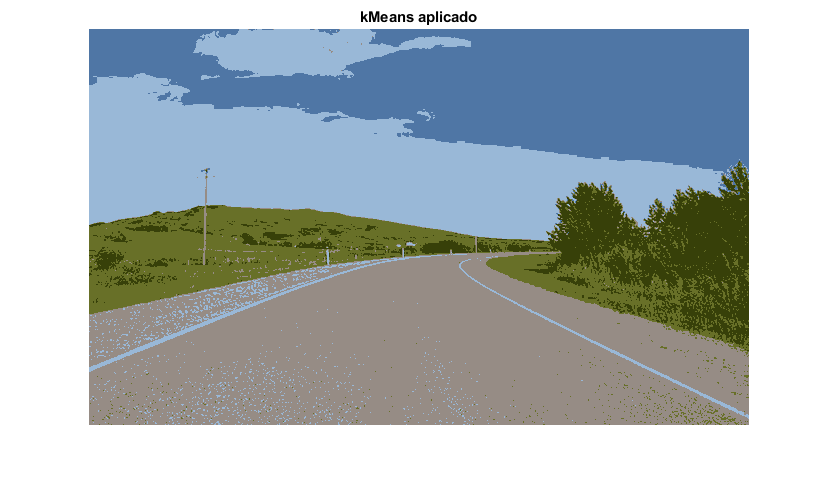


Ilustración 15. Filtrado con , primera corrida.



Ilustración 16. Filtrado con , segunda corrida.



Ilustración 17. Filtrado con .

# A.3.2 Análisis:

Con el uso de kMeans se ha podido simplificar la imagen original a varias imágenes filtradas, especialmente se ha podido observar que mientras mayor sea el valor de k, mayor es el detalle del resultado de la imagen, reconociendo asimismo que se parecería más a la imagen original y que, a la vez es menos comprimida. Finalmente, se pudo ver que el uso del filtrado por moda lo que afecta a la imagen original es en el de hacerla más borrosa.