K-Means

Prof. Juan Samuel Pérez

Mayo, 2021

Asignación 1

Realice el siguiente ejercicio:

1. Simule 5 fuentes de datos que generan 300 muestras bidimensionales cada una. Los valores de cada dimension para cada fuente son generados como variables aleatorias con distribución Gaussiana $\mathcal{X}_n \sim \mathcal{N}(\mu_n, \sigma_n^2)$, para $n = 1, ..., 10$. Los valores de μ_n y σ_n para cada dimensión de cada fuente deben ser generados como variables aleatorias con distribución uniforme según $\mu_n \sim U(0, 100)$ y $\sigma_n \sim U(0, 15)$. Utilice el mismo dataset para todos los experimentos que siguen.

2. Aplique K-means para agrupar estos datos utilizando $K = 5$. Utilice gráficas apropiadas en que compare el resultado final del proceso de *clustering* con los grupos de las fuentes de datos originales. Guarde un registro de cuántas iteraciones fueron necesarias para que el algoritmo lograra convergencia. Represente, además, la posición de los centroides finales. No es necesario graficar los resultados de cada iteración de K-means en sus resultados.

3. Repita el *clustering* 3 veces, utilizando diferentes inicializaciones para los centroides en cada corrida. Compare y analice los resultados.

Asignación 2

Emplear K-Means para realizar segmentación por colores de una imagen:

Ejecute K-Means para K=3,4,5,6 sobre la imagen 'highway.jpg' provista. Sustituya los píxeles de la imagen original por los centroides de los clústers encontrados a los que pertenecen. Aplique un filtro de moda de tamaño adecuado (por ejemplo 9x9) a la imagen resultante para homogeneizar los valores de las regiones encontradas. Presente por separado los segmentos de la imagen original encontrados usando técnicas de enmascaramiento (*masking*).

Pista: Las funciones colormap() y modefilt() de MATLAB pueden ser útiles.
Restricción: Queda totalmente prohibido utilizar la función kmeans() nativa de MATLAB/Python.