



Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI5438 – Inteligencia Artificial II
Enero-Marzo 2017

Leonardo Martínez Carné: 11-10567

Nicolás Mañán Carné: 06-39883

Joel Rivas Carné: 11-10866

Proyecto 3 – Clustering

En el siguiente proyecto se realizarán 3 ejercicios enfocados en el uso e implementación del algoritmo de agrupamiento (*clustering*), *k-means*.

Ejercicio 1:

La implementación del algoritmo K-means se encuentra en el archivo *kmeans.py* dentro de los archivos del proyecto.

Ejercicio 2:

Se realizó la clasificación con 2, 3, 4 y 5 clusters para el conjunto de datos Iris-setosa. Para cada conjunto, se muestra que resultado se obtuvo para los datos originales.

2 Clusters

Clase	Iris Setosa	Iris Versicolor	Iris Virginica
0	50	4	0
1	0	46	50

En este caso, vemos que para Iris Setosa la clasificación fue correcta fue de un 100% separandola en una de las clases (0). Mientras que en el caso de Iris Versicolor e Iris Virginica se tuvieron algunos falsos positivos en la primera, clasificando en la case 1.

3 Clusters

Clase	Iris Setosa	Iris Versicolor	Iris Virginica
0	0	12	49
1	0	38	1
2	50	0	0

Para 3 clusters, vemos que para Iris Setosa la clasificación fue correcta nuevamente un 100% separándola en este caso en la clase 2. Mientras que en el caso de Iris Versicolor e Iris Virginica se puede ver que se separaron en mayor parte en las clases 2 y 1, respectivamente. Obteniendo algunos falsos positivos.

4 Clusters

Clase	Iris Setosa	Iris Versicolor	Iris Virginica
0	50	0	0
1	0	3	40
2	0	41	10
3	0	6	0

Para 4 clusters, Iris setosa fue clasificada en 100% para la clase 0. Mientras que se nota que el algoritmo clasificó en su mayoría las en las clases 1 y 2 a Iris Virginica e Iris Versicolor respectivamente. Para la clase 3 se colocaron algunos elementos de Iris Versicolor.

5 Clusters

Clase	Iris Setosa	Iris Versicolor	Iris Virginica
0	0	46	14
1	0	1	35
2	0	3	1
3	0	0	0
4	50	0	0

Usando 5 clusters, se obtiene nuevamente un 100% para Iris Setosa en la clase 4, mientras que para Iris Versicolor e Iris Virginica el resultado fue mixto. La mayoría de la primera en la clase 0 y la mayoría de la segunda en la clase 1.

En general, las opciones que dan mejor resultado respecto a la clasificación son para 2 y 3 clases, en todos los casos se observa que para Iris-Setosa la clasificación es siempre exacta a partir de 3 clases mientras que para Iris Versicolor e Iris Virginica poseen mayor número de falsos positivos/negativos.

Ejercicio 3:

Se utilizó la librería PIL (*Python Image Library*) para leer los pixeles de las imágenes a trabajar y guardarlos en listas para que el algoritmo *k-mean* los reciba y realice la compresión utilizando $K = \{2,4,8,16,32,64,128\}$, grupos.

La compresión de imágenes está dada por reducir el número de colores que contiene la imagen, se calcularan 'k' colores, dados anteriormente, como los centroides de los grupos (*clusters*) y se reemplazará cada pixel en la imagen con el color del centroide más cercano.

Se utilizaron varias imágenes de distintas resoluciones, sin embargo, el tiempo de ejecución para imágenes de más de 300 x 300 pixeles era demasiado extenso para ser un ejemplo factible.

En el archivo comprimido se encontrarán tres carpetas de imágenes con las que se trabajó, la imagen de un zorro de 100x100 pixeles, el ojo de una mujer de 200x100 pixeles y un ojo humano de 600x600 pixeles con resultados hasta $K = 32$. Estas carpetas se encuentran dentro del directorio *data/E3*. En el caso del ojo de la mujer, este corresponde a un fragmento de una imagen más grande. Se muestra tanto la imagen original, como la imagen con el ojo reemplazado por la imagen resultante del algoritmo con $k = 128$.

Para tratar de mejorar el numero de iteraciones antes de la convergencia, se probó una version alternativa del ejercicio 3 pero inicializando aleatoriamente solo la mitad de los centroides, utilizando los centroides resultantes de las iteraciones anteriores. Se logró reducir el número de iteraciones para $k > 16$, obteniendo una compresión de la imagen en calidad bastante similar. En el directorio *data/E3/Zorro2* se puede ver la compresión con esta técnica, a modo de comparación con la imagen del Zorro de 100x100 pixeles.