Práctica 7. Clustering con K-Means y Mezcla de Gaussianas

Objetivo

El objetivo es utilizar técnicas de agrupamiento basadas en **K-Means** y **Mezcla de Gaussianas** (**GMM**), entender su funcionamiento y utilizarlas para resolver un problema de compresión de imágenes.

Estudio previo

- Repasa las transparencias de clase y estudia las funciones auxiliares proporcionadas para esta práctica.
- Revisa la documentación de scikit-learn de las clases `KMeans` y `GaussianMixture`.
- Revisa **BIC** como criterio para selección de modelos. En esta práctica será útil para la selección de número de componentes en *GMM*.

Desarrollo de la práctica

1. **Datos simulados**: vamos a comenzar con datos simulados de una mezcla de Gaussianas. Sigue el guión del notebook para:
   a) Entender la generación de datos simulados.
   b) Comparar los resultados de *k-means* y *GMM* con el número correcto de componentes.
   c) Seleccionar el número de componentes (clústers) en base al criterio *BIC* (ver método `.bic`) en *GMM* y al valor de la función de *distorsión* en *k-means* (ver método `.score`).
2. **Agrupamiento en el espacio de RGB de una imagen**. En esta segunda parte, vamos a usar *k-means* y *GMM* para clusterizar, y por tanto, cuantizar, los colores de una imagen. En primer lugar, haremos clustering sobre la imagen, ya aportada, `parrotcolors.jpg`. Aprovechando lo aprendido en el apartado anterior:
   a) Visualiza y analiza la distribución 3D (a clusterizar) de puntos RGB de `parrotcolors.jpg`.
   b) Utiliza *k-means* y *GMM* para clusterizar, con un número de clústers / componentes a tu elección, el espacio RGB. Con *GMM*, emplea `"tied"` en el argumento `covariance_type` y usa la variable, definida en el notebook, `init_inv_cov` como valor de `precisions_init`.
   c) Selecciona el número de clústers/componentes "`k`" usando la misma métrica/criterio que antes. Como feedback adicional, visualiza las imágenes cuantizadas para cada `k`, es decir, usando para cada píxel la media del clúster al que se ha asociado. Compara y razona los resultados en ambos casos.
   d) *Opcional*: Estudia en más profundidad el modelo GMM. Para ello, puedes ver el efecto de diferentes `covariance_type` (s) y explicar cómo afectan a los resultados. Tendrás que ajustar la inicialización de las covarianzas inversas (`precisions_init`). También puedes variar otros parámetros, como la inicialización del modelo (`init_params`). En este caso puede que tengas que proponer una alternativa.

3. Con los resultados y el análisis del punto anterior, elige dos imágenes donde esta técnica de compresión funcione muy bien y muy mal, respectivamente. Piensa en el ratio de compresión que se va a conseguir en función del número de clusters. Asume que la imagen original tiene tres bytes por pixel (valores RGB), mientras que la comprimida solo debe guardar las medias de los clusters y a qué cluster pertenece cada píxel.

A entregar en Moodle

Un notebook `P7.ipynb` con el código de cada apartado, los resultados, su interpretación y las conclusiones que hayas obtenido.