

## Tarea 5 EL4106 - Semestre Otoño 2021

Profesor: Javier Ruiz del Solar  
Auxiliar: Patricio Loncomilla

Fecha enunciado: Jueves 03 de junio de 2021  
Plazo entrega tarea: Miércoles 16 de junio de 2021

El objetivo de esta tarea es utilizar distintos algoritmos de clustering y analizar su desempeño. Se utilizará la base de datos Anuran Calls MFCCs, que es una base de datos tomadas del UC Irvine Machine Learning Repository. Contiene llamadas de 10 especies distintas de rana, las cuales son representadas mediante 22 características. Dichas características son coeficientes cepstrales en frecuencias de *Mel*. La escala *Mel* es una transformación de frecuencias (en Hertz) a un espacio nuevo (en *Mel*) en la cual las alturas de los sonidos están perceptualmente equiespaciadas. La base de datos se puede bajar desde la siguiente dirección:

<https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+%28MFCCs%29>

Para poder realizar el trabajo, se usará el siguiente código base (basado en scikit-learn):

[https://scikit-learn.org/stable/auto\\_examples/cluster/plot\\_kmeans\\_digits.html#sphx-glr-auto-examples-cluster-plot-kmeans-digits-py](https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html#sphx-glr-auto-examples-cluster-plot-kmeans-digits-py)

**Se pide utilizar tres algoritmos de clustering en este trabajo: k-means, DBSCAN y clustering aglomerativo.**

- 1) Implemente un código que lea la base de datos. Se recomienda usar la biblioteca Pandas. Los datos se deben dividir en características (MFCCs\_1 – MFCCs\_22) y labels (Species). Los labels deben transformarse a números en el rango 0-9.
- 2) Modifique la función `bench_k_means()` agregando una columna extra, correspondiente al número de clusters.
- 3) Implemente una nueva función `bench_clustering2()`, tomando como base la función `bench_k_means()` modificada en el punto anterior. Esta función será usada para obtener un benchmark tanto de DBSCAN como del clustering aglomerativo. Notar que:
  - a. DBSCAN no posee la variable `inertia_`, esto debe ser manejado por el alumno.
  - b. DBSCAN no asigna todas las muestras a un cluster, las muestras no asociadas tienen label "-1". Debido a esto, la función `bench_DBSCAN()` debe entregar la opción de: (i) calcular las métricas sólo con los datos asociados, o (ii) calcular las métricas usando todos los datos, asumiendo que los datos no asociados conforman un cluster extra. Dicha opción debe ser implementada usando una variable extra (True/False) que se le entrega a la función.
  - c. El algoritmo DBSCAN no genera una cantidad predeterminada de clusters.
  - d. Dados algunos parámetros, es posible que DBSCAN no encuentre clusters, o bien encuentre uno solo, lo cual imposibilita calcular las métricas. Decida una estrategia para manejar estos casos.
- 4) Hacer pruebas con distintas variantes de algoritmos:
  - a. K-Means (con inicialización al azar)
  - b. K-means++
  - c. DBSCAN con `epsilon` por defecto
  - d. DBSCAN con `epsilon` 0.7
  - e. DBSCAN con `epsilon` 0.2
  - f. DBSCAN con `epsilon` por defecto, agregando outliers a cluster extra
  - g. DBSCAN con `epsilon` 0.7, agregando outliers a cluster extra
  - h. DBSCAN con `epsilon` 0.2, agregando outliers a cluster extra
  - i. Clustering aglomerativo

- j. Repetir todas las pruebas anteriores, después de usar PCA sobre los datos para reducirlos a 2 dimensiones.
- 5) Analice los resultados obtenidos:
- a. Compare los algoritmos aplicados en base a cuatro métricas (*completeness*, *homogeneity*, *v-measure*, *silhouette*), indicando: (i) cuál variante es mejor según cada métrica y (ii) cuál variante es peor según cada métrica. Se debe considerar tanto los datos originales como los casos en los cuales se aplicó PCA.
  - b. Indique en cuáles casos no fue posible calcular las métricas de DBSCAN. Analice el número de clusters obtenidos por DBSCAN y su efecto sobre las métricas. Analice el efecto de considerar o no los outliers de DBSCAN como un cluster extra y su efecto sobre las métricas.
  - c. Analice el efecto de usar PCA sobre las métricas obtenidas.

Los informes y códigos deben ser subidos a u-cursos a más tardar el día Miércoles 16 de junio a las 23:59. Incluir un corto archivo de texto explicando cómo se utiliza su programa. Las tareas atrasadas serán penalizadas con un punto base por cada día de atraso, incluyendo fines de semana.

**Importante:** La evaluación de la tarea considerará el correcto funcionamiento del programa, la inclusión de los resultados de los pasos pedidos en el informe, la calidad de los experimentos realizados y de su análisis, la inclusión de las partes importantes del código en el informe, así como la forma, prolijidad y calidad de este. En u-cursos hay un documento indicando la estructura esperada para los informes.

Recordatorio: se debe subir los informes a la plataforma turnitin

#### Referencias

1. <https://archive.ics.uci.edu/ml/datasets/Anuran+Calls+%28MFCCs%29>
2. <https://scikit-learn.org/stable/modules/clustering.html>