

README del repositorio

En el fichero `requirements.txt` se encuentran listadas todas las librerías utilizadas, para facilitar su instalación.

Algoritmos

Los algoritmos de clustering están implementados en los siguientes ficheros:

- `kmeans.py` (la función principal ejecuta K-Means si se usa la distancia euclidea, y K-Medoids en otro caso)
- `agglomerative.py` (método aglomerativo)
- `fuzzy.py` (Fuzzy C-Means)
- `dbscan.py` y `dbscan_vptree.py` (DBSCAN implementado a fuerza bruta y con un VP-árbol, respectivamente)
- `EM.py` (*Expectation-Maximization*)

Para todos estos ficheros, salvo `dbscan_vptree.py`, existen scripts con nombre `main_nombredefichero.py` que contienen casos de prueba para los algoritmos.

Métricas

En `metrics.py` se encuentran implementadas las métricas de evaluación de *clustering*: Índice de Silueta, Calinski-Harabasz, Davies-Bouldin, índice C y el Índice de Dunn.

Estimación del número óptimo de clusters

En `optimal_n_clusters.py` se encuentran métodos para estimar el número de clusters, entre ellos el método del codo, una función para graficar valores de métricas utilizando K-Means con distintos valores de K, y el *Gap statistic*.

Funciones auxiliares

En `experiment_functions.py` están las funciones auxiliares utilizadas para la realización de experimentos.

Experimentos

- `wines_experiment.py`: experimento con el conjunto de vinos. Ejecutándolo con el argumento 1 se guarda la tabla de resultados usando el dataset original en la imagen `results_wines_raw.svg`, y con el argumento 2 se guarda la tabla correspondiente al dataset estandarizado mediante Z-score en `results_wines_stand.svg`.
- `customers_experiment.py`: experimento con el *dataset* de clientes. Ejecutando el programa, se guarda la tabla de resultados del experimento en la imagen `results_customers.svg`.