

PRÁCTICA 3: ALGORITMOS K-MEDIAS, BAYES Y LLOYD.

- Lenguaje de programación utilizado: He decidido programar el algoritmo en el lenguaje C++ ya que este me brinda la libertad, robustez, flexibilidad y potencia que consideraba necesaria para programar el algoritmo.
- Procedimiento de implementación:
 1. El programa ha sido creado subdividiéndolo en varios módulos que unidos constituirán la aplicación. Estos módulos son:
 - main.cpp: Contiene el punto de arranque de la aplicación, su misión consiste en llamar al menú principal.
 - menuPrincipal.h: Su labor se limita a pintar por pantalla el menú con el que interactuará el usuario para que este decida como quiere utilizar el programa. El usuario podrá elegir entre varias opciones:
 1. Cargar datos para estimación paramétrica de Bayes.
 2. Introducir una muestra para clasificación Bayesiana.
 3. Cargar datos para K-Medias difuso.
 4. Recalcular centros para K-Medias difuso.
 5. Introducir una muestra para clasificación con K-Medias difuso.
 6. Cargar datos para Lloyd.
 7. Introducir una muestra para clasificarla con Lloyd.
 8. Salir del programa.
 - funcionesMenuPrincipal.h: Este módulo sirve para desacoplar el código fuente del menú principal que no se dedica a su representación, es decir, contiene las funciones que serán invocadas desde el menú principal para ejecutar la funcionalidad requerida por el usuario.
 - clase.h: Contiene el objeto con el que se trabaja cuando escogemos el algoritmo de Bayes. Contiene los atributos indicados en el algoritmo, tales como el vector media, el nombre de la clase a la que representa, la matriz de covarianza de la clase, sus submatrices de covarianza, la matriz de covarianza inversa y los métodos necesarios para el cálculo de estos. De entre ellos podemos destacar la distancia de Mahalanobis o el método de Gauss para el cálculo de matrices inversas.

- representante.h: Contiene el objeto con el que se trabaja cuando escogemos el algoritmo K-Medias. Contiene los atributos requeridos por el algoritmo, tales como el nombre de la clase a la que representa, su vector central actual y su vector central anterior. También posee el valor de la tolerancia, el peso exponencial y una colección de muestras asociadas a la clase. Estos tres últimos atributos serán visibles por todos los representantes de las clases que existan. Este objeto también tiene los métodos necesarios para implementar los cálculos del algoritmo, de entre ellos podemos destacar el cálculo del grado de pertenencia de una muestra al representante de la clase o el cálculo de los centros en función de los grados de pertenencia.
- centro.h: Contiene el objeto con el que se trabaja cuando escogemos el algoritmo de Lloyd. Contiene los atributos indicados en el algoritmo, tales como el nombre del centro de la clase a la que representa, el centro actual y el centro anterior. También posee el valor de la tolerancia, el máximo de iteraciones que el algoritmo debe realizar y la razón de aprendizaje. Estos tres últimos atributos serán visibles por todos los centros de las clases que existan. Este objeto también tiene los métodos necesarios para implementar los cálculos del algoritmo, de entre ellos podemos destacar como se recalcula el valor de el centro en función de una muestra dada.
- funciones.h: Este módulo contiene otras funciones cuya funcionalidad estará requerida por las tres clases anteriores.
- funcionesDeCarga.h: Es el módulo que contiene las funciones necesarias para realizar la carga de los datos en función del algoritmo a utilizar.
- operadores.h: En este módulo se redefine el comportamiento de los operadores aritméticos básicos para el tipo vector y matriz, de forma que estos definan como se realizan dichas operaciones para estos elementos.
- comparadores.h: Contiene los objetos comparadores necesarios para que las estructuras de datos actúen (ordenen los datos) en base al comportamiento requerido por los algoritmos.
- tipos.h: Módulo en el que se redefinen los tipos básicos de datos del lenguaje C++ para hacerlos más legibles durante la lectura de código, particularmente, para facilitar la legibilidad de cada uno de los tres algoritmos implementados.

2. Estructuras de datos que he considerado necesarias para la implementación del algoritmo:
 - Vectores.
 - Diccionarios.
 - Colas de prioridad.
- Parte básica realizada:
 1. Clasificador de Bayes.
- Ampliaciones realizadas:
 1. Algoritmo K-Medias.
 2. Algoritmo de Lloyd.