Algoritmo Borroso

Este algoritmo recibe los datos de las clases y los datos de los centros iniciales.

El algoritmo ejecuta una función performIteration() que devuelve un booleano que indica si es necesario seguir iterando según el parámetro épsilon que nos da la práctica.

Lo primero que hace el método performIteration() es calcular la distancias. Para ello iteramos sobre la matrix dataMatrix que contiene los datos de las 2 clases. Dentro del bucle llamamos al método calculateDistancesBetween(xArray, vArray) y que calcula dxx siguiendo la fórmula descrita en los apuntes, página 7. El valor obtenido lo almacenamos en una matrix denominada, distances.

A continuación calculamos los grados de pertenencia. Para ello en un bucle iteramos sobre la matrix dataMatrix que contiene los datos de las 2 clases. Este bucle llama al método calculateMembershipgrade(int, int, Matrix). Este método aplica la ecuación (2) de las tansparencias de la teoría. El resultado obtenido lo almacenamos en la matrix membershipGradesMatrix.

A continuación recalculamos los centros en un siguiente bucle y lo almacenamos en la variable initialCentroMatrix.

Finalmente iteramos sobre los centros, obtenemos su distancia, y comparamos con épsilon. Si la distancia es mayor o igual que épsilon, hemos terminado las iteraciones, en caso contrario el método devolverá false indicado que necesita iterar de nuevo para hallar la solución.

Para obtener el resultado de un caso de prueba, utilizamos el método getClassForValues(doublé[]). Este método itera sobre los valores y aplica el algoritmo Borroso con los datos anteriormente calculados. Por último construye un string con los datos obtenidos de la ejecución del algoritmo.