

K Means clustering

Objetivos

- Trabajar el concepto de Tipo Abstracto de Datos.
- Aprender a utilizar el tipo `std::valarray<>`.

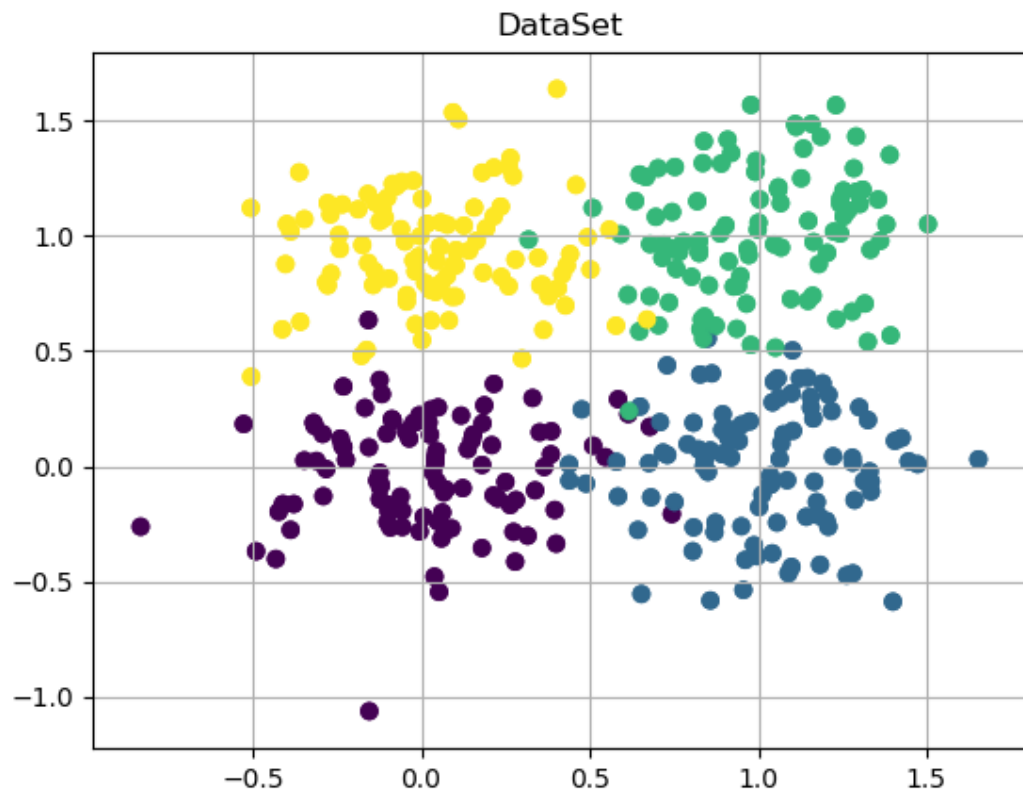
Descripción

Diseñar un TAD que modele el concepto de “patrón” usado en Aprendizaje Automático. Vamos a simplificar asumiendo que un patrón consiste en un vector en un espacio R^n (usaremos flotantes de simple precisión) junto con un valor entero que se considerará como etiqueta de pertenencia a una clase. Se asumirá que un valor negativo de etiqueta significa que el patrón no está adscrito a ninguna clase.

El TAD Pattern deberá ofrecer una interfaz con operaciones para crear/consultar/modificar las propiedades del patrón. Así mismo se ofrecerá un conjunto de operaciones matemáticas comunes (véase la interfaz descrita en `pattern.hpp` en los ficheros asociados). Para la implementación se espera que se utilice como base el tipo `std::valarray<>` (ver referencia [2]) utilizando al máximo todas las operaciones vectoriales que proporciona.

Usando el TAD Pattern, deberás codificar un programa que aplique el algoritmo K-MEANS [1] a un conjunto de patrones cargado desde un fichero. Este algoritmo buscará en un conjunto de patrones los K mejores agrupamientos (“cluster”).

Se probará el código utilizando el siguiente DataSet generado de manera aleatoria con 4 clusters centrados en los puntos (0,0), (1,0), (1,1) y (0,1).



Un ejemplo ejecución sería:

```
Iterations: 12
Centroids:
0 0.984984 0.993965
1 1.00086 -0.00572748
2 -0.00333882 -0.0334698
3 0.031067 0.948197
```

Como el algoritmo no es determinista, cada ejecución dará resultados distintos, pero se espera que para cada uno de los centroides verdaderos (0,0), (1,0), (1,1) y (0,1) exista un centroide calculado con distancia euclídea < 0.1 .

Referencias.

- [1] https://en.wikipedia.org/wiki/K-means_clustering
- [2] <http://www.cplusplus.com/reference/valarray/valarray/>