

Grupowanie

1 k -means

1. Losowo wybierz k centroidów.
2. Powtarzaj do momentu konwergencji (kiedy przydziały do grup pozostaną niezmiennie w dwóch kolejnych iteracjach):
 - (a) Dla każdego przykładu znajdź najbliższy centroid i przypisz go do odpowiadającej mu grupy.
 - (b) Dla każdej grupy wylicz nowy centroid, będący średnią arytmetyczną wektorów w grupie.

2 Hierarchiczne grupowanie aglomeracyjne

1. Zaczynij z każdym przykładem w swojej własnej grupie.
2. Znajdź dwie najbliższe grupy zgodnie z przyjętą metryką i połącz je.
3. Powtarzaj krok 2 do momentu, kiedy wszystkie przykłady będą w jednej grupie.

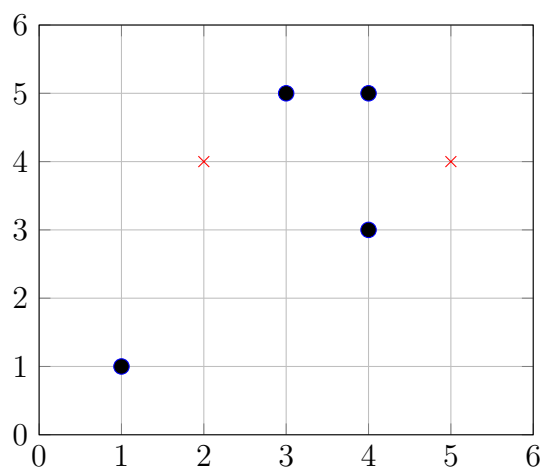
Zadania

Zadanie 1.

Grupuj następujące zbiory algorytmem k -means:

- (a) A(1, 1), B(3, 5), C(4, 3), D(4, 5).

Początkowe centroidy: $c_1(2, 4)$, $c_2(5, 4)$



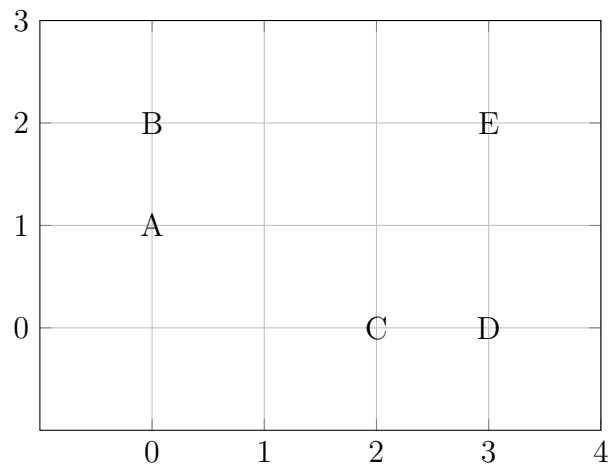
- (b) A(4, 8, 4, 3), B(4, 6, 3, 6), C(2, 8, 9, 3), D(7, 7, 2, 3).

Początkowe centroidy: $c_1(4, 8, 4, 3)$, $c_2(2, 8, 9, 3)$

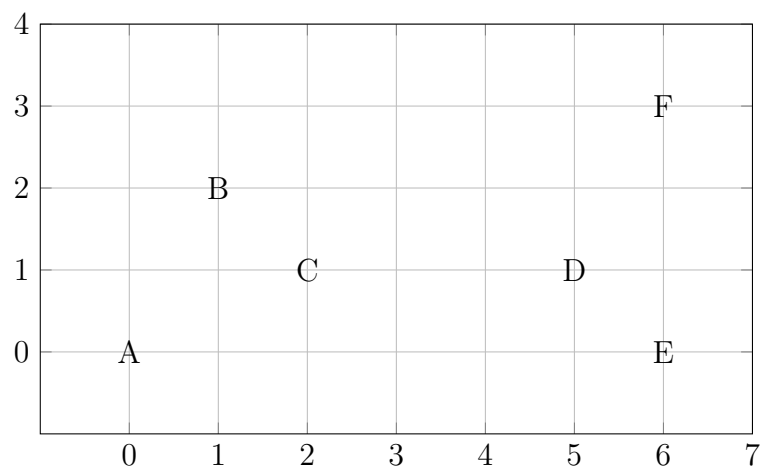
Zadanie 2.

Grupuj następujące zbiory metodą hierarchicznego grupowania aglomeracyjnego. Narysuj dendrogramy.

(a) $A(0,1)$, $B(0,2)$, $C(2,0)$, $D(3,0)$, $E(3,2)$.



(b) $A(0,0)$, $B(1,2)$, $C(2,1)$, $D(5,1)$, $E(6,0)$, $F(6,3)$.



Mini-projekt: k -means

Implementuj algorytm k -means. Grupuj dane z pliku `iris.data`.

Uwaga: Grupowanie to zadanie uczenia bez nadzoru, więc należy zignorować atrybut decyzyjny. Można wykorzystać go pod koniec do ewaluacji grupowania.

Program powinien:

- Umożliwiać wybór k .
- Po każdej iteracji: wypisywać sumę odległości przykładów od ich centroidów. Ta wartość powinna zmniejszać się z każdą iteracją. Uwaga: należy wypisywać sumę dla wszystkich przykładów, a nie każdej grupy osobno. Przykład:
Iteracja 1: 126.38
Iteracja 2: 86.34
Iteracja 3: 49.91
...
- Na końcu: wyświetlać składy grup.
- Dodatkowo (opcjonalnie): wyświetlać miary czystości grup, np. procentowe zawartości każdej z klas Iris, lub entropię.