Programowanie sieciowe

Klasteryzacja danych - algorytm Kohonena - Lab4

Data:	19.06.2022	Dzień:	Wtorek TN + 1/2
Grupa:	Y02-15b	Godzina:	17:05
Numer indeksu:	252889	Prowadzący:	dr inż. Marek Bazan
Nazwisko i imię: Nowek Jakub			

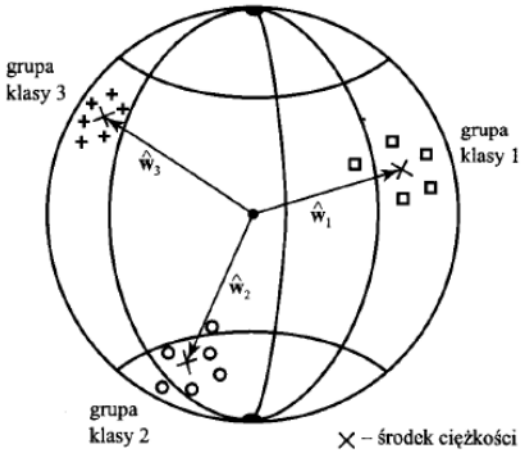
Spis treści

1. Opis problemu

Należało zaimplementować algorytm Kohenena, który na podstawie danych wejściowych oraz zadeklarowanej liczby klas wyznacza wektory, wskazujące środek ciężkości klas - zadanie klasteryzacji. Do sprawdzenia działania algorytmu wykorzystane zostały dane Iris oraz dane giełdowe spółki KIM. Poniżej umieszczono link do bazy danych zawierającej dane spółek, która została wykorzystana.
`https://www.kaggle.com/datasets/camnugent/sandp500/metadata`

2. Opis użytych algorytmów

Zadaniem algorytmu Kohenena jest wskazanie środków ciężkości zbiorów danych, należących do określonej liczby klas.



Rys. 1: Pożądany wynik działania algorytmu Kohonena dla zbioru danych posiadających 3 klasy.

Zaznaczone na rysunku wektory $\hat{w_1}$, $\hat{w_2}$, $\hat{w_3}$ to poszukiwane wektory reprezentantów. Ich liczba jest równa liczbie klas, która ostała zadeklarowana na wejściu algorytmu. Jeśli pojedynczy punkt danych charakteryzuje l właściwości, to każdy wektor reprezentantów będzie l-wymiarowy.

Na początku działania algorytmu inicjowane są wektory reprezentantów, jako wektory współliniowe, leżące na jednej prostej, z tym samym punktem zaczepienia. W trakcie działania algorytmu położenie wektorów zmienia się. Każdy kolejny analizowany punkt wpływa na kierunek tego wektora, którego koniec znajduje się najbliżej niego.

Aktualizowanie wektora dokonywane jest na podstawie jednej z trzech wybranych miar:

$$\hat{w}_m^T \hat{x} = \max_{i=1,2,\ldots,p} \hat{w}_i^T \hat{x} \tag{1}$$

$$\left|\hat{w}_m^T - \hat{x}\right| = \min_{i=1,2,\ldots,p} \left|\hat{w}_i^T - \hat{x}\right| \tag{2}$$

$$\left|\hat{w}_m^T - \hat{x}\right| = \min_{i=1,2,\ldots,p} \sqrt{\sum_{j=1}^{l} \left|\hat{w}_{ij}^T - \hat{x}_j\right|} \tag{3}$$

Po wybraniu najlepszej wartości miary, wektor dla którego była ona optymalna, aktualizowany jest przy pomocy współczynnika uczenia α, gdzie k stanowi numer iteracji.

$$\hat{w}_m^{(k+1)} = \hat{w}_m^{(k)} + \alpha^{(k)} \left(\hat{x} - \hat{w}_m^{(k)}\right) \tag{4}$$

Współczynnik α może być w każdej iteracji modyfikowany na trzy sposoby. T jest maksymalną liczbą iteracji po zbiorze.

1. Liniowe zmniejszanie
$$\alpha^{(k)} = \alpha^{(0)}(T-k)/T, \quad k = 1, 2, \ldots, T \tag{5}$$
2. Wykładnicze zmniejszanie
$$\alpha^{(k)} = a^{(0)} \exp(-Ck), \quad k = 1, 2, \ldots, T, \quad C > 0, \qquad \text{C - pewna stała} \tag{6}$$
3. Hiperboliczne zmniejszanie
$$\alpha^{(k)} = C1/\left(C2 + k\right), \quad k = 1, 2, \ldots, T, \quad C1, C2 > 0, \qquad \text{C1, C2 - pewne stałe} \tag{7}$$

3. Testy numeryczne

3.1. Definicja testów

Testy zostały przeprowadzone na zbiorze Iris oraz na zbiorze danych giełdowych firmy KIM.

- Przetestowane zostało działanie algorytmu dla danych znormalizowanych i danych bez normalizacji.
- Sprawdzono działanie w zależności od użytej miary oraz od sposobu modyfikacji współczynnika uczenia α.
- Sprawdzono działanie algorytmu dla wartości początkowych współczynnika uczenia ze zbioru $\{0.1, 0.3, 0.7\}$.

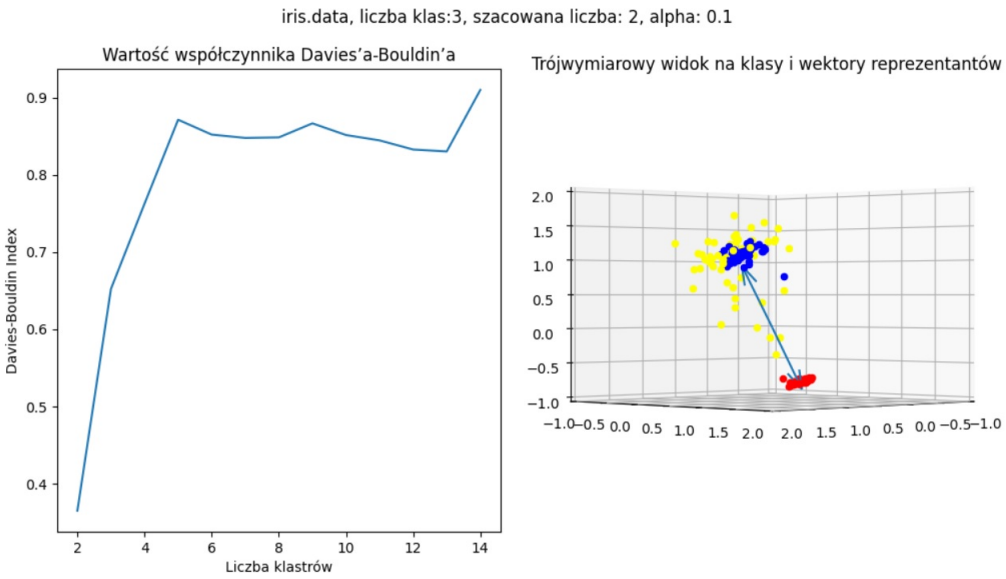
Wyniki zaprezentowano w postaci wykresów trójwymiarowych. Na koniec, przy pomocy algorytmu Davies'a-Bouldin'a oszacowano rzeczywistą liczbę klastrów.

4. Wyniki.

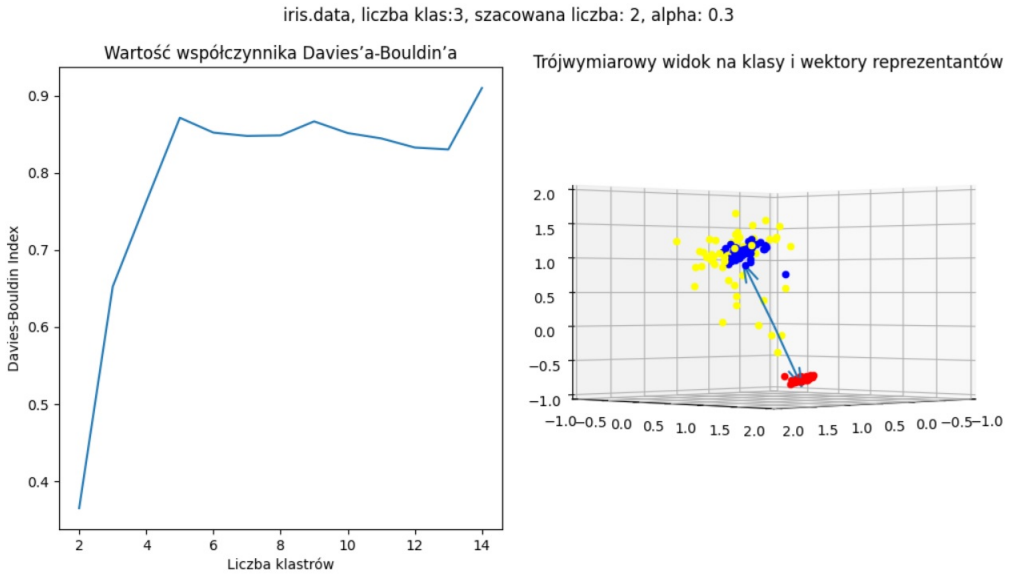
Wyniki testów zaprezentowano na poniższych rysunkach.

4.1. Wpływ wartości α.

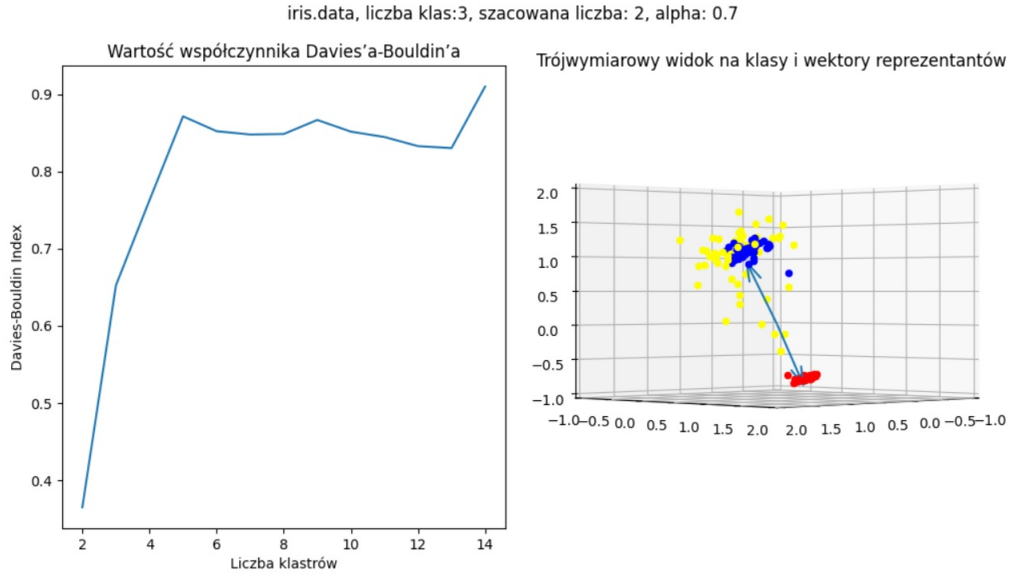
- Norma pierwsza
- Liniowe zmniejszanie współczynnika uczenia α
- dane Iris



Rys. 2: $\alpha = 0.1$



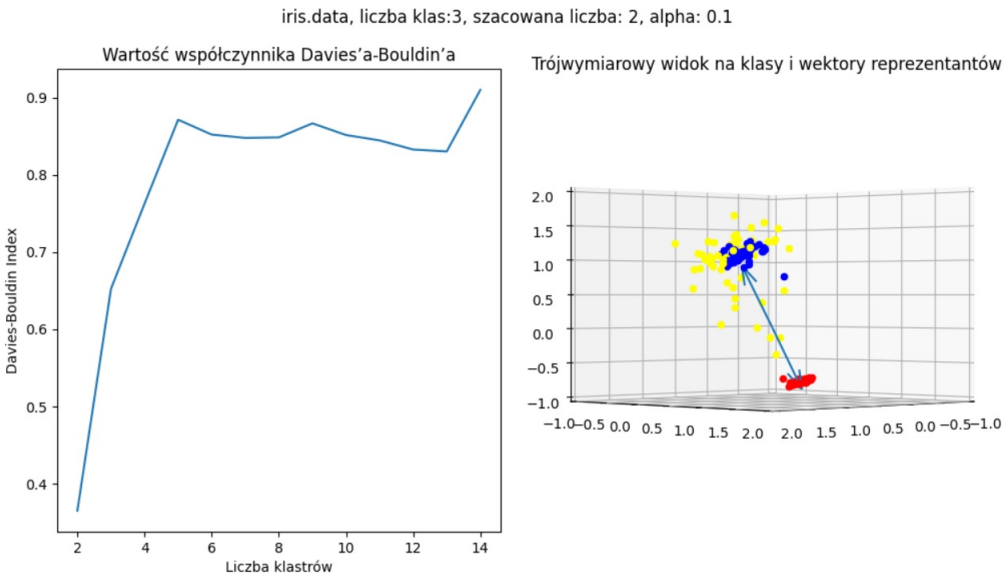
Rys. 3: $\alpha = 0.3$



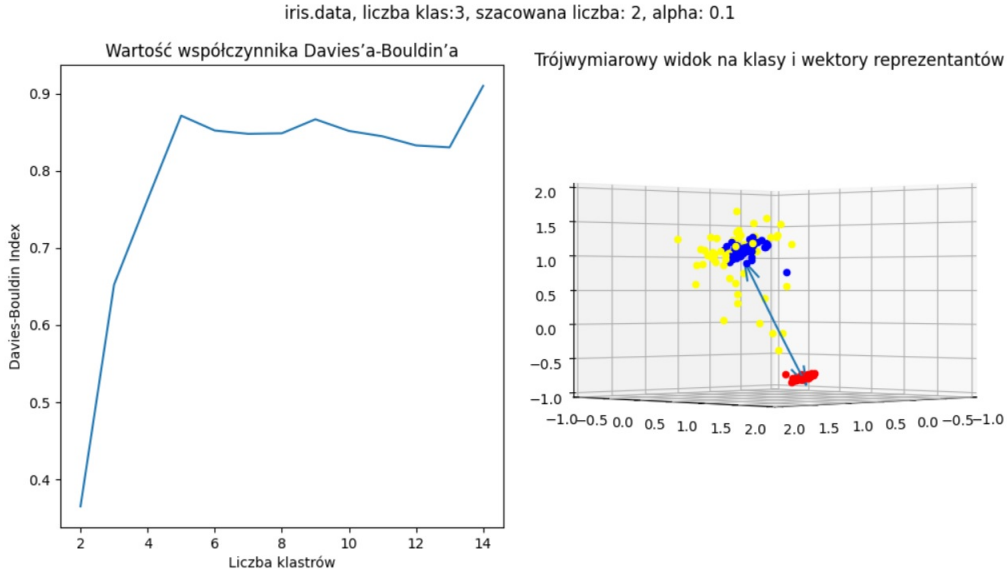
Rys. 4: $\alpha = 0.7$

4.2. Wpływ metody modyfikacji współczynnika α.

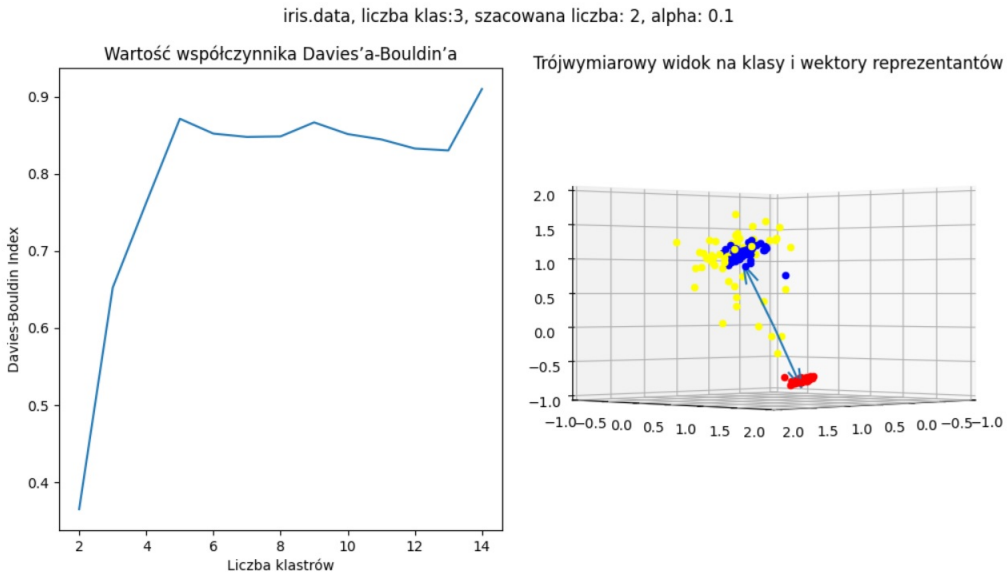
- Norma pierwsza
- $\alpha_0 = 0.1$
- dane Iris



Rys. 5: Liniowe zmniejszanie



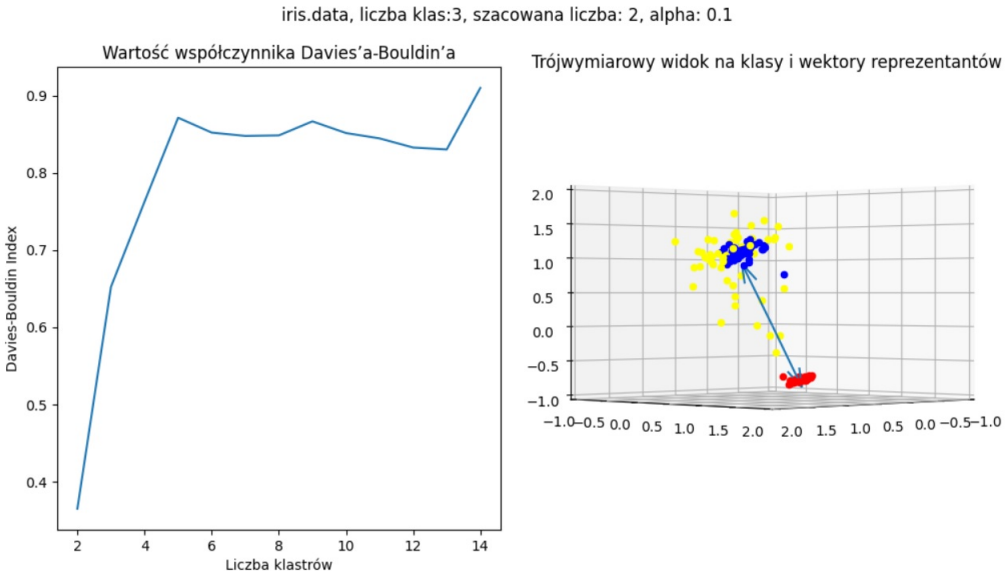
Rys. 6: Wykładnicze zmniejszanie



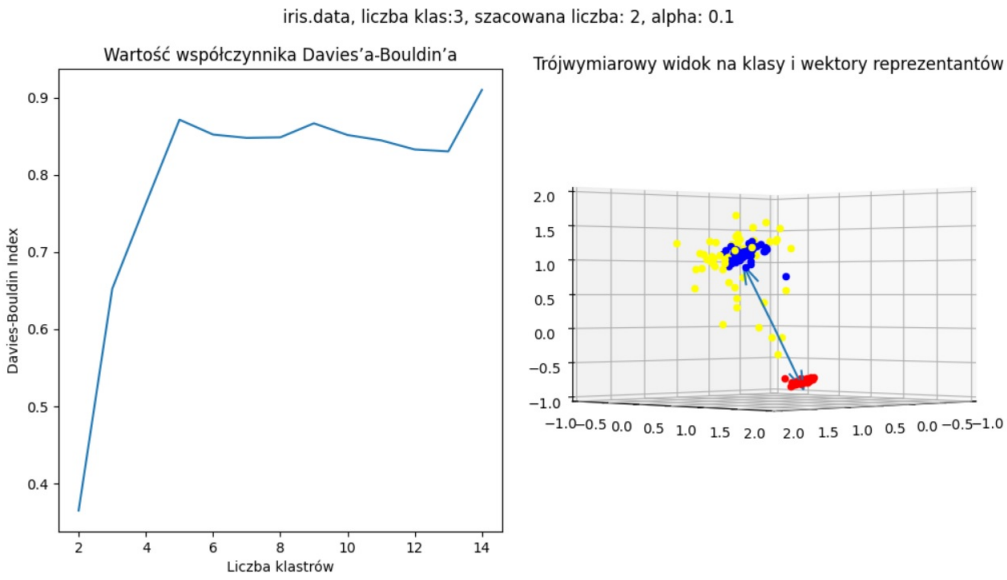
Rys. 7: Hiperboliczne zmniejszanie

4.3. Wpływ normy użytej do określania optymalnego wektora reprezentantów.

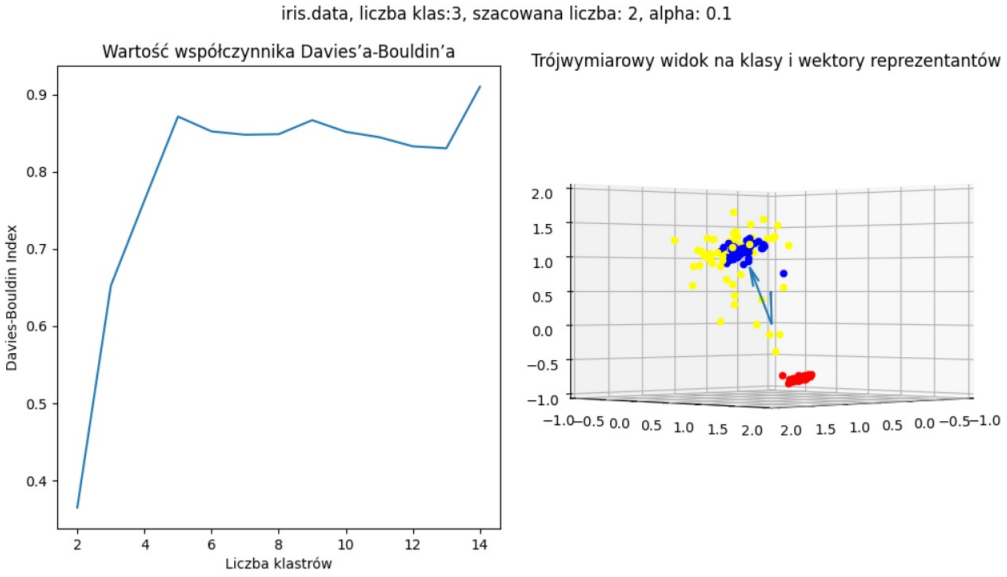
- Liniowe zmniejszanie współczynnika uczenia α
- $\alpha_0 = 0.1$
- dane Iris



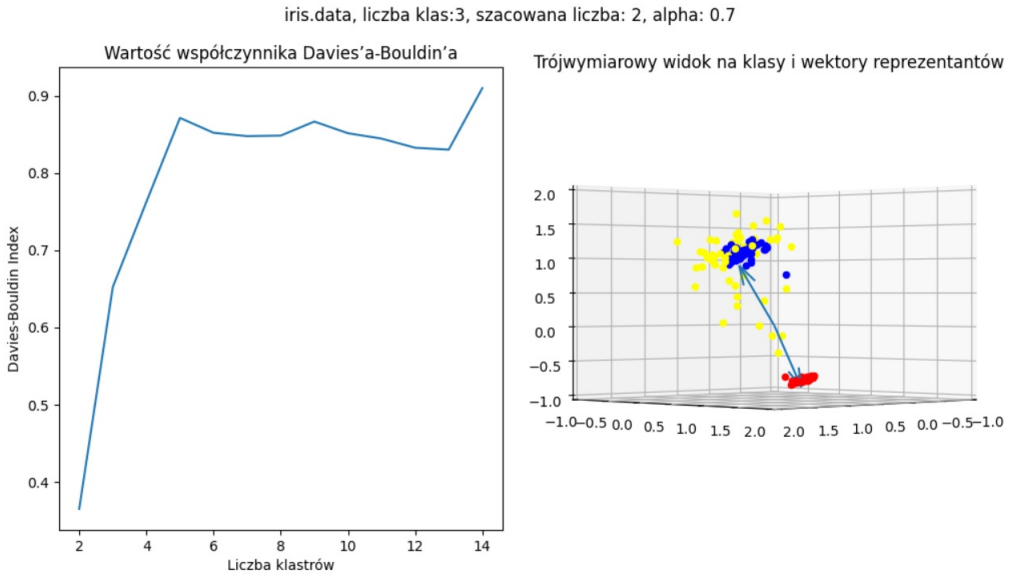
Rys. 8: Norma pierwsza



Rys. 9: Norma druga



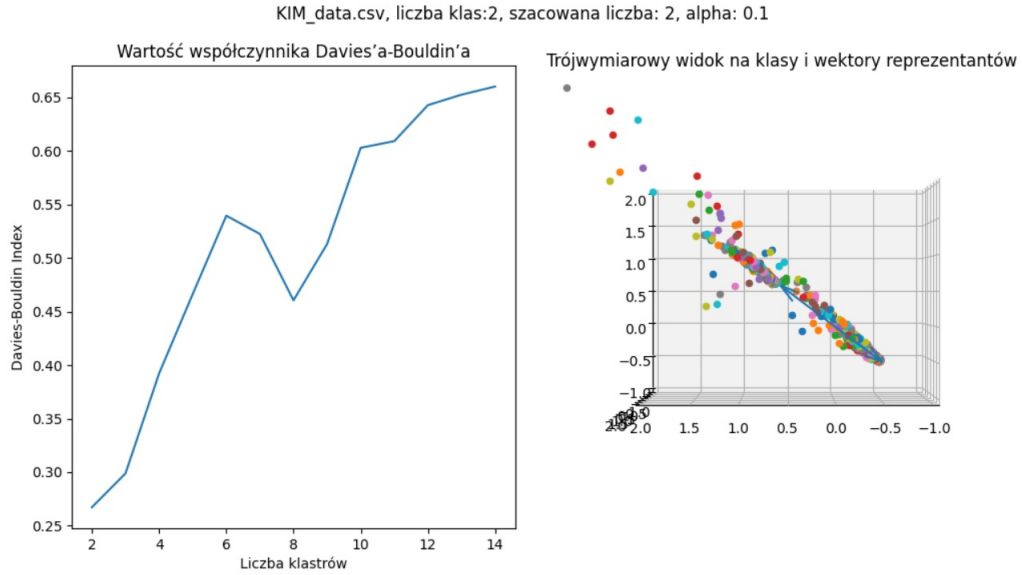
Rys. 10: Norma trzecia, α_0=0.1



Rys. 11: Norma trzecia, α_0=0.7

4.4. Dane giełdowe.

- Liniowe zmniejszanie współczynnika uczenia α
- $\alpha_0 = 0.1$
- dane giełdowe KIM
- Norma pierwsza



Rys. 12: Dane giełdowe KIM - znormalizowane

5. Wnioski.

- W przeciwieństwie do poprzednio używanych algorytmów, algorytm Kohonena, jako algorytm uczenia bez nadzoru do działania potrzebuje jedynie zbioru testowego - nie wymaga konstruowania zbioru treningowego, zawierającego wyjścia.
- Ponieważ w zbiorze Iris dwie z trzech klas nachodzą na siebie, algorytm Kohonena nie jest w stanie ich poprawnie rozróżnić. Algorytm Davies'a-Bouldin'a zastosowany do wykrycia liczby klas również oszacował liczbę klas w zbiorze Iris na 2.
- Ponieważ dane w zbiorze Iris nie mają charakteru rosnącego lub malejącego, tylko oscylują w pewnych stałych granicach, algorytm Kohonena może działać na nieznormalizowanych danych.
- Po znormalizowaniu danych, wpływ zmiany parametrów na wyniki jest niezbyt widoczny. Zmiany poszczególnych parametrów powodują niewielkie przesunięcia, bądź obrócenia wektorów reprezentantów.
- Najbardziej widoczny był wpływ zmiany normy użytej do określania optymalnego wektora reprezentantów. Norma Manhattan (norma trzecia) okazała się odnosić skutek dla początkowej wartości $\alpha_0 = 0.7$, w przeciwieństwie do dwóch pozostałych norm, które lepiej radzą sobie przy małych wartościach początkowych współczynnika uczenia.
- Zbiór danych giełdowych ma charakter rosnący co uniemożliwia badania przy nieznormalizowanych danych wejściowych.
- Zaprezentowanie wyników przy wymiarze pojedynczego punktu danych większym od 3 wymaga użycia rzutowań.