

DBSCAN

Projektowanie systemów i sieci komputerowych

Mateusz Śliwa

Piotr Świder

Spis treści

[1. Wprowadzenie. 3](#_Toc182248244)

[2. Charakterystyka. 3](#_Toc182248245)

[**2.1.** **Parametry.** 3](#_Toc182248246)

[**2.2.** **Główne założenia.** 3](#_Toc182248247)

[**2.3.** **Złożoność.** 3](#_Toc182248248)

[3. Podsumowanie. 3](#_Toc182248249)

[**3.1.** **Zalety.** 4](#_Toc182248250)

[**3.2.** **Wady.** 4](#_Toc182248251)

[**3.3.** **Zastosowania.** 4](#_Toc182248252)

# Wprowadzenie.

Algorytm DBSCAN (ang. Density-based spatial clustering of applications with noise) jest algorytmem klasteryzacji danych opartym na gęstości. Został on wymyślony w 1996 roku przez Martina Estera. Klastry utworzone za pomocą algorytmu charakteryzują się dużym zagęszczeniem punktów w stosunku do otoczenia. Algorytm umożliwia tworzenie klastrów o dowolnej wielkości oraz kształcie.

# Charakterystyka.

## **Parametry.**

Aby poprawnie opisać algorytm DBCAN niezbędne będzie użycie dwóch parametrów.

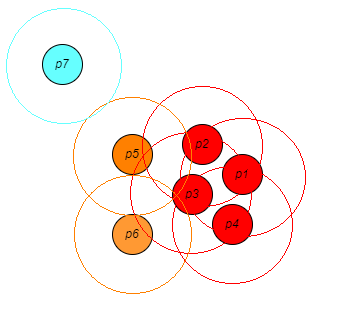
* *ε (eps)* będzie określać maksymalny promień sąsiedztwa między punktami.
* *minPts* oznaczający minimalną liczbę punktów (obiektów) w klastrze.

## **Główne założenia.**

Zakładając pewny zbiór *X* punktów (obiektów), oraz dwa parametry z poprzedniego podpunktu, można utworzyć odpowiedni klaster. Należy jednak wcześniej wprowadzić pewne oznaczenia oraz nazewnictwo punktów, mianowicie: punkt centralny, punkt osiągalny (szczególnie punkt bezpośrednio osiągalny), punkt szumu (*noise*).

* Punkt *p* *∈  X*  jest punktem centralnym, gdy w promieniu *ε* od punktu *p*  znajduje się minPts punktów (wliczając w to punkt *p.*)
* Punkt *q ∈  X*  jest punktem bezpośrednio osiągalnym gdy znajduje się on w odległości *ε* od punktu *p.*
* Punkt *q ∈  X* jest punktem osiągalnym gdy istnieje ścieżka prowadząca między punktami *p* oraz *q*.
* Punkty szumu, są to wszystkie pozostałe nieosiągalne punkty.

Zbiór wszystkich punktów osiągalnych oraz punktu p tworzą odpowiedni klaster, tak jak na poniższym przykładowym schemacie:



Zakładając, że *p1* jest punktem centralnym:

* *p2, p3, p4* są punktami bezpośrednio osiągalnymi
* *p5, p6* są punktami osiągalnymi
* *p7* jest punktem szumu

Utworzony klaster: *p1, p2, p3, p4, p5, p6*

## **Złożoność.**

Ogólna średnia złożoność algorytmu DBSCAN wynosi O(*n log n*), a w najgorszym wypadku (przykładowo gdy wszystkie punkty znajdują się od siebie w większej odległości niż *ε*) wynosi ona O(*n2*).

# Podsumowanie.

DBSCAN jest niezwykle użytecznym algorytmem, który swoje zastosowania odnalazł w wielu dziedzinach, jak choćby analiza danych czy też uczenie maszynowe. Jednak jak każde narzędzie posiada on swoje wady, zalety oraz praktyczne zastosowania.

## **Zalety.**

DBCSAN nie potrzebuje wcześniejszego określenia wielkości klastra ani ilości klastrów co umożliwia dynamiczne ich tworzenie. Dodatkowo wykrywa szum (*noise*). Dzięki temu, że wymaga jedynie dwóch parametrów jest on stosunkowo prosty w zrozumieniu oraz implementacji.

## **Wady.**

Algorytm DBSCAN nie jest tak optymalny w przypadku wysoce zróżnicowanych danych, gdzie także ciężko może być ustalić i wybrać odpowiednie wartości parametrów. Użyteczność i jego zastosowanie zależy w dużej mierze od wybrania parametru *ε*.

## **Zastosowania.**

Z racji na specyficzne cechy algorytmu DBSCAN posiada on zróżnicowane zastosowania.

* Dzięki możliwości wykrywania punktów szumu algorytmu DBSCAN używa się do wykrywania anomalii w danych.
* DBSCAN można zastosować w marketingu do identyfikacji klastrów klientów o podobnych preferencjach.
* W wypadku uczenia maszynowego, algorytmu DBSCAN używa się do wizualizacji oraz znajdywania pewnych wzorców oraz struktur.
* Algorytm DBSCAN jest używany w analizie przestrzennej do identyfikacji podobnych regionów lub regionów o zbliżonych właściwościach.