

Uczenie nienadzorowane

algorytmy grupowania wykład 12

Jerzy Stefanowski

Instytut Informatyki PP

2021

Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI-TECH)
projekt finansowany z środków Programu Operacyjnego Polska Cyfrowa
POPC.03.02.00-00-0001/20



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Plan wykładu

- Rozszerzenia klasycznych algorytmów grupowania
 - Algorytm k-średnich
 - K-medoid, PAM, ...
 - Algorytmy hierarchiczne
 - BIRCH
- Algorytmy gęstościowe
 - DBSCAN

----- druga część wykładu -----

- Podejścia wykorzystujące modele statystyczne
 - Algorytm mieszanin rozkładów (EM)
- Ocena jakości grupowania
- Podsumowanie

Przypomnienie podziału metod

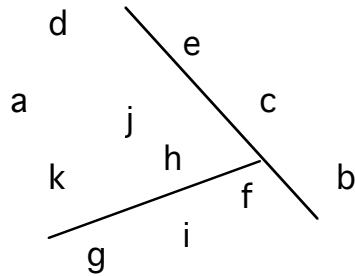
- Podziałowo-optymalizacyjne: Znajdź podział na zadaną liczbę skupień wg. zadanego kryterium.
- Metody hierarchiczne: Zbuduj drzewiastą strukturę skupień.
- Gęstościowo (Density-based): Poszukuj obszarów o większej gęstości występowania obserwacji
- Grid-based: wykorzystujące wielowymiarowy podział przestrzeni siatką ograniczeń
- Model-based: hipoteza co do własności modelu pewnego skupienia i procedura jego estymacji.

Czym jest skupienie?

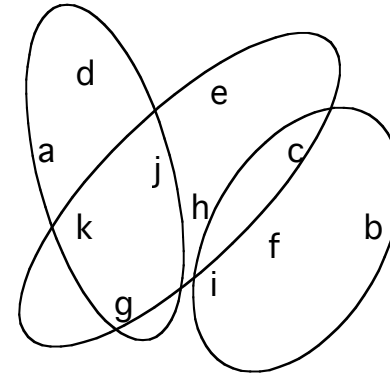
1. Zbiorem najbardziej podobnych obiektów
2. Podzbiór obiektów, dla których odległość jest mniejsza niż ich odległość od obiektów z innych skupień.
3. Podobszar wielowymiarowej przestrzeni zawierający odpowiednio dużą gęstość obiektów, oddzielony od innych podobszarów o dużej gęstości strefą rzadkiego występowania obiektów

Różne sposoby reprezentowania skupisk

(a)



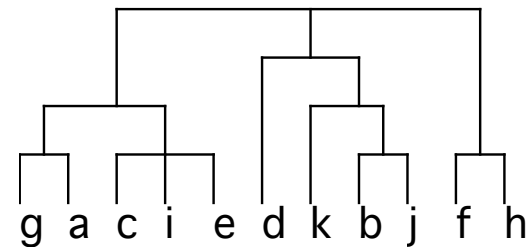
(b)



(c)

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1
...			

(d)



Algorytmy podziałowo – optymalizacyjne

- Zadanie: Podzielenie zbioru obserwacji na K zbiorów elementów (skupień C), które są jak najbardziej jednorodne
- Jednorodność – funkcja oceny
- Intuicja \rightarrow zmienność wewnątrzskupieniowa $wc(C)$ i zmienność międzyskupieniowa $bc(C)$

Możliwe są różne sposoby zdefiniowania

- np. wybierzmy środki skupień \mathbf{r}_k (centroidy) $\mathbf{r}_k = \frac{1}{n_k} \sum_{\mathbf{x} \in C_k} \mathbf{x}$
- Co prowadzi do

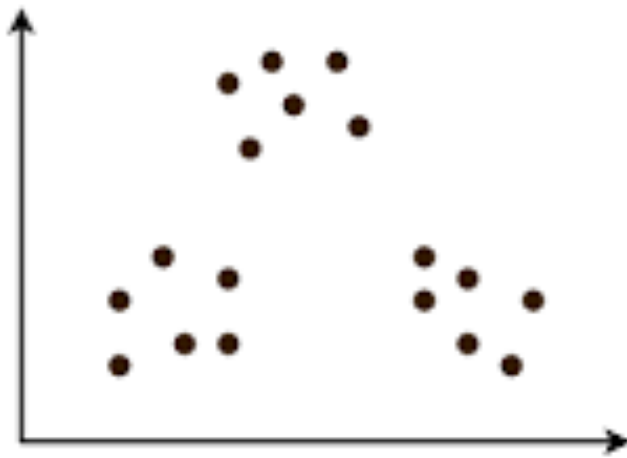
$$wc(C) = \sum_{k=1}^K \sum_{\mathbf{x} \in C_k} d(\mathbf{x}, \mathbf{r}_k)^2$$

$$bc(C) = \sum_{1 \leq j < k \leq K} d(\mathbf{r}_j, \mathbf{r}_k)^2$$

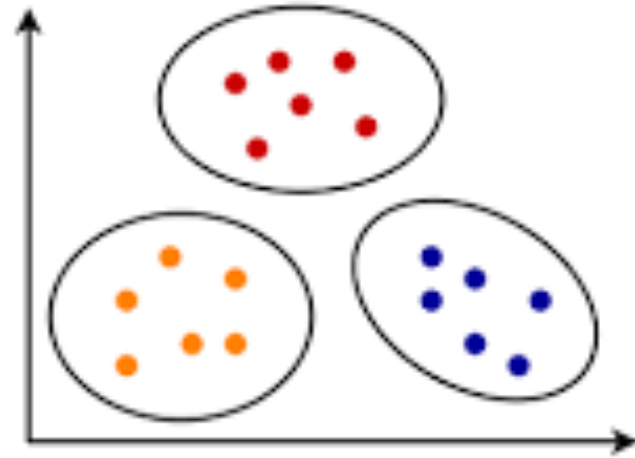
Algorytm k średnich (k – means)

- Cel: k -średnich \rightarrow minimalizacja $wc(C)$
- Przeszukiwanie przestrzeni możliwych przypisań \rightarrow bardzo kosztowne (oszacowanie w ks. Koronackiego)
- Problem optymalizacji kombinatorycznej \rightarrow systematyczne przeszukiwanie metodą iteracyjnego udoskonalania:
 - Rozpocznij od rozwiązania początkowego (losowego).
 - Ponownie przypisz punkty do skupień tak, aby otrzymać największą zmianę w funkcji oceny.
 - Przelicz zaktualizowane środki skupień, ...
 - Postępuj aż do momentu, w którym nie ma już żadnych zmian w funkcji oceny lub w składzie grup.
- Zachłanne przeszukiwanie \rightarrow proste i prowadzi do co najmniej lokalnego minimum. Różne modyfikacje, np. rozpoczynania od kilku rozwiązań startowych
- Złożoność algorytmu K - średnich $\rightarrow O(Knl)$

Ilustracja k-średnich

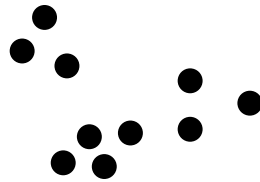
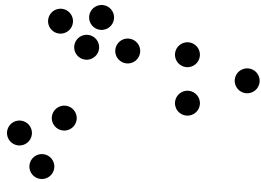


Before K-Means

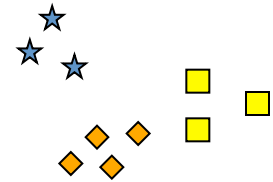
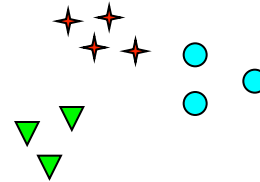


After K-Means

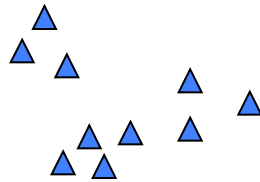
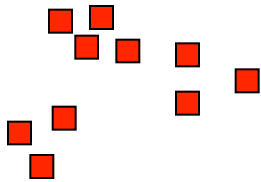
Trudność określenia liczby skupisk, ...



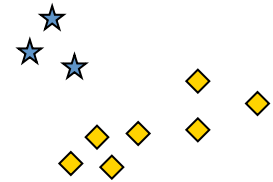
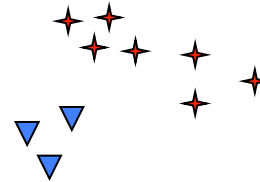
Ile jest naturalnych skupisk?



Sześć skupisk



Dwa skupiska



Cztery skupiska

Ustalanie liczby skupień

Liczbę skupień wybiera się na podstawie przesłanek merytorycznych albo szacuje się je metodami hierarchicznymi. Można dokonać obliczeń dla wszystkich wartości k z ustalonego przedziału:

$$k_{\min} \leq k \leq k_{\max}$$

Możliwe są różne podejścia:

1. Wybór kryterium oceny skupisk
2. Uruchomienie algorytmu k-średnich dla różnych k
3. Propozycja podziału dla najlepszego k

Alternatywnie: - użyj innego algorytmu (np. hierarchicznego) do identyfikacji możliwej liczby skupisk na podstawie dendrogramu

Niezależnie:

K-średnich dość czułe na obecność obserwacji samotniczych (ang. outliers) – mogą tworzyć pojedyncze skupiska i zakłócać grupowanie pozostałych przykładów -> warto wykryć i „odłożyć” ze zbioru do niezależnej

Preferencja dla tworzenia sferycznych kształtów skupisk

Dobór k w algorytmie k -średnich

- X-means popularne w implementacjach (WEKA, Python)
- Stosowane kryteria oceny podziału na skupiska:
 - Bayesian Information Criterion (BIC)
 - Akaike Information Criterion (AIC)
- Operacja “improve structure” – podział wybranego skupiska na dwa.
- D. Pelleg and A. Moore (2000) X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In Proceedings of the 17th International Conf. on Machine Learning, 727--734.

X-means

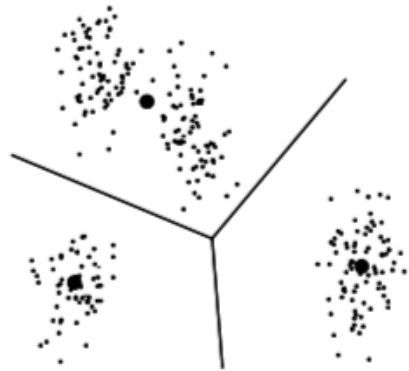


Figure 1. The result of running K-means with three centroids.

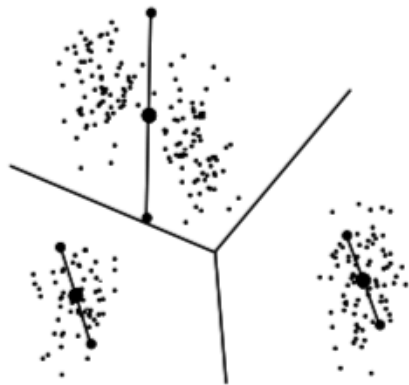
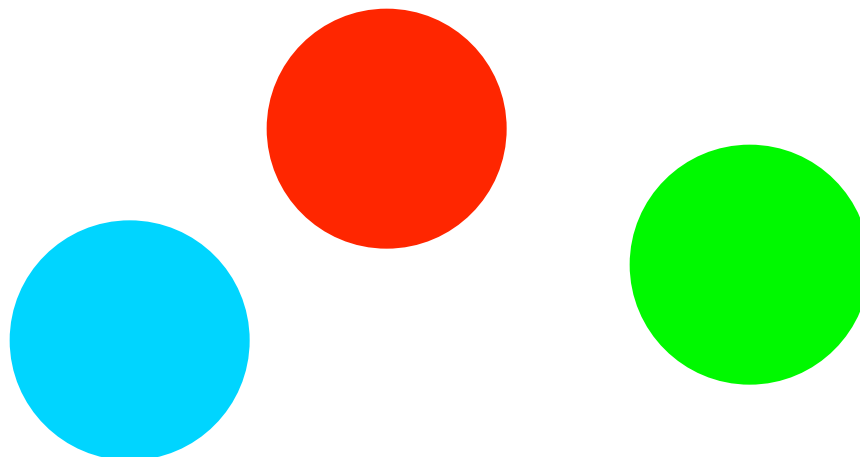


Figure 2. Each original centroid splits into two children.

Pewne ukierunkowanie K-średnich

- Tworzy się „kuliste” kształty skupień



- Co z obserwacjami odstającymi i nieregularnymi kształtami skupień?

K-means krótkie podsumowanie

Zalety

- Proste i łatwe do zrozumienia
- Reprezentacja skupień jako centroidy

Wady

- Jawne podanie liczby skupień
- Wszystkie przykłady muszą być przydzielone do skupień
- Problem z outliers (za duża wrażliwość)
- Ukierunkowanie na jednorodne „sferyczne” kształty skupień

Dalsze rozszerzenia k -średnich

- Rozmyte k -means (Fuzzy ISODATA)
- Wersja k -medoids
- Rozszerzenia dla przetwarzania dużych wolumenów danych, np. PAM
- Inspiracje dla modeli statystycznych (EM)
- Odniesienia do grupowania spektralnego

Obszerne omówienie w pracy:

Warto zapoznać się z książką S.Wierzchoń, M.Kłopotek:
Algorytmy analizy skupień. WNT 2015

Inne spojrzenie na skupiska

Crisp vs. soft clusters (ang. fuzzy clustering)

Obiekt x_i może należeć do wielu skupisk C_j w różnym stopniu przynależności z zakresu $[0;1]$

Najbardziej znany algorytm Fuzzy c-mean [Bezdek]

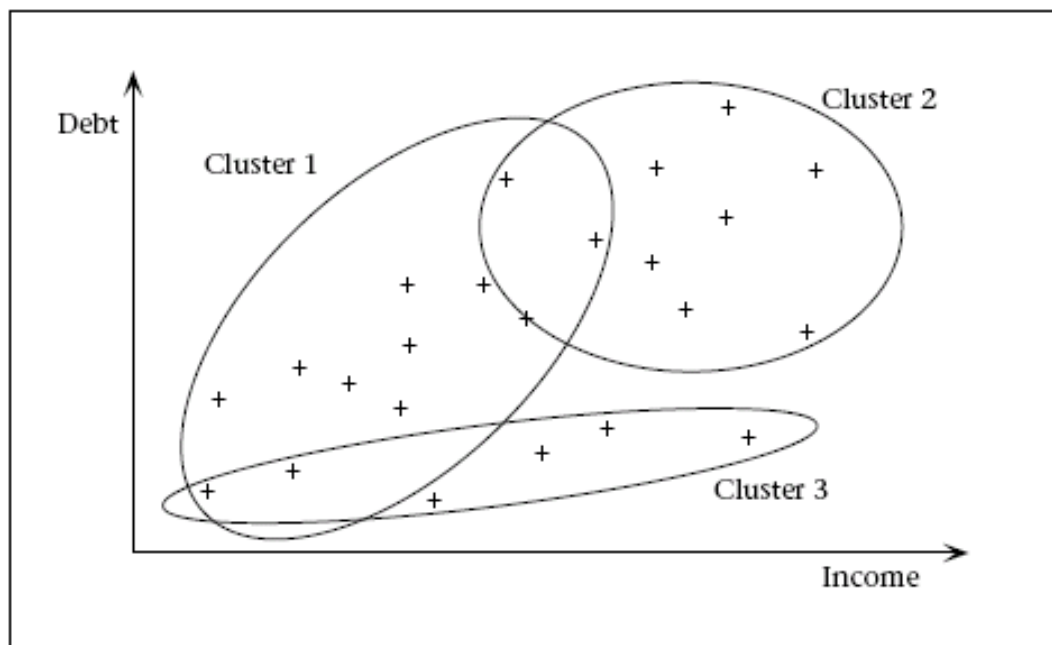


Figure 5. A Simple Clustering of the Loan Data Set into Three Clusters.

Note that original labels are replaced by a +.

Algorytm k -medoids

- Ograniczenie standardowego k -średnich, m.in.:
 - Czuły na obserwacje odstające i tzw. szum
 - Konieczność przeliczania macierzy odległości w każdej iteracji
- Algorytm k -medoids (PAM)
 - Zastąpienie reprezentanta skupiska - średniego obiektu (na ogół sztuczne położonego) poprzez rzeczywisty obiekt z danych położony najbliżej centrum
 - Zmiany w algorytmie – inny sposób oceny wymiany obiektów (**medoidów**) w kolejnych iteracjach -> najbardziej znana wersja PAM (Partitioning Around Medoid) - Kaufman i Rousseeuw 1987

Przykład wpływu outliers

- Dla standardowego k-średnich i jednego atrybutu x rozważ skupisko:
 - średnia z obserwacji o wartościach 1,3,5,7,9 wynosi 5
 - Jeśli ostatnia obserwacja ulegnie zmianie na 1,3,5,7, 1009 to średnia będzie 205
- Dla k-medoid (obiekt najbliższy centrum) – dla 1,3,5,7, 1009 będzie to 7, a w kolejnych iteracjach przesunie się na 5

PAM – algorytm k-medoids

1. Wybierz (losowo) k rzeczywistych obiektów jako załączki skupisk
 2. Przydziel każdy obiekt do tego skupiska, gdzie jest najbliższy medoid
 3. W kolejnym kroku aktualizuje się położenia centroidów – medoidów i powtarzany jest przydział obiektów do najbliższego skupiska
- Wymiana obiektów z medoidami na podstawie oszacowanie specjalnej funkcji kosztu wymiany (SWAP) – uwaga analizuje się wszystkie pary (obiekt vs. medoid)
 - Zaproponowano specjalne funkcje niepodobieństwa dla atrybutów jakościowych

K-medoids przykład

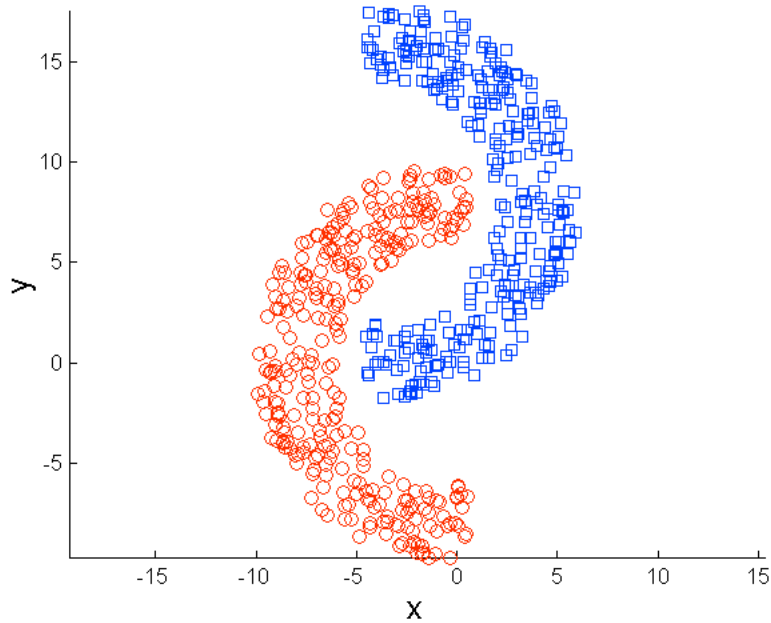
- Można przeanalizować prosty przykład dwuwymiarowy opisany na blogu <https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/>
- Lub podręczniki J.Han i inni Data Mining.

PAM – ograniczenia dla zbyt masywnych danych

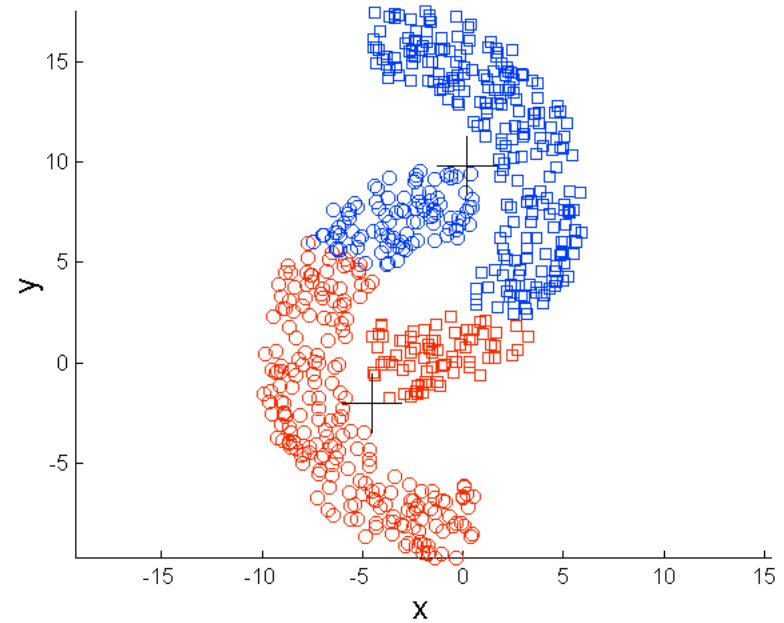
- Wada PAM – słaba skalowalność dla relatywnie dużych wolumenów danych
 - Ocena złożoność PAM $O(k(n-k)^2)$, gdzie n – liczba obiektów, k – liczba skupisk
- **CLARA** (ang. Clustering Large Applications) – próba poprawy skalowalności
 - Losowana reprezentatywna próba danych z całych danych
 - PAM poszukuje zbioru dobrych medoidów
 - Jeśli dobrze dobrana próba, to będą także odzwierciedlać rozkład całych danych
 - Możliwe powtórzenie losowania i wybór najlepszego zbioru medoidów
 - Przydział wszystkich obiektów do skupisk wg. wybranych medoidów
- Inne próby modyfikacji dla przyspieszenia obliczeń kosztów wymiany $O(k)$ – Schubert i in. Fast and eager k-medoids clustering 2020.

Ograniczenia k-mean i motywacje dla innych algorytmów

Ograniczenia K-średnich: Niesferyczne kształty



**Oryginalny zestaw
danych**



K-means (2 skupienia)

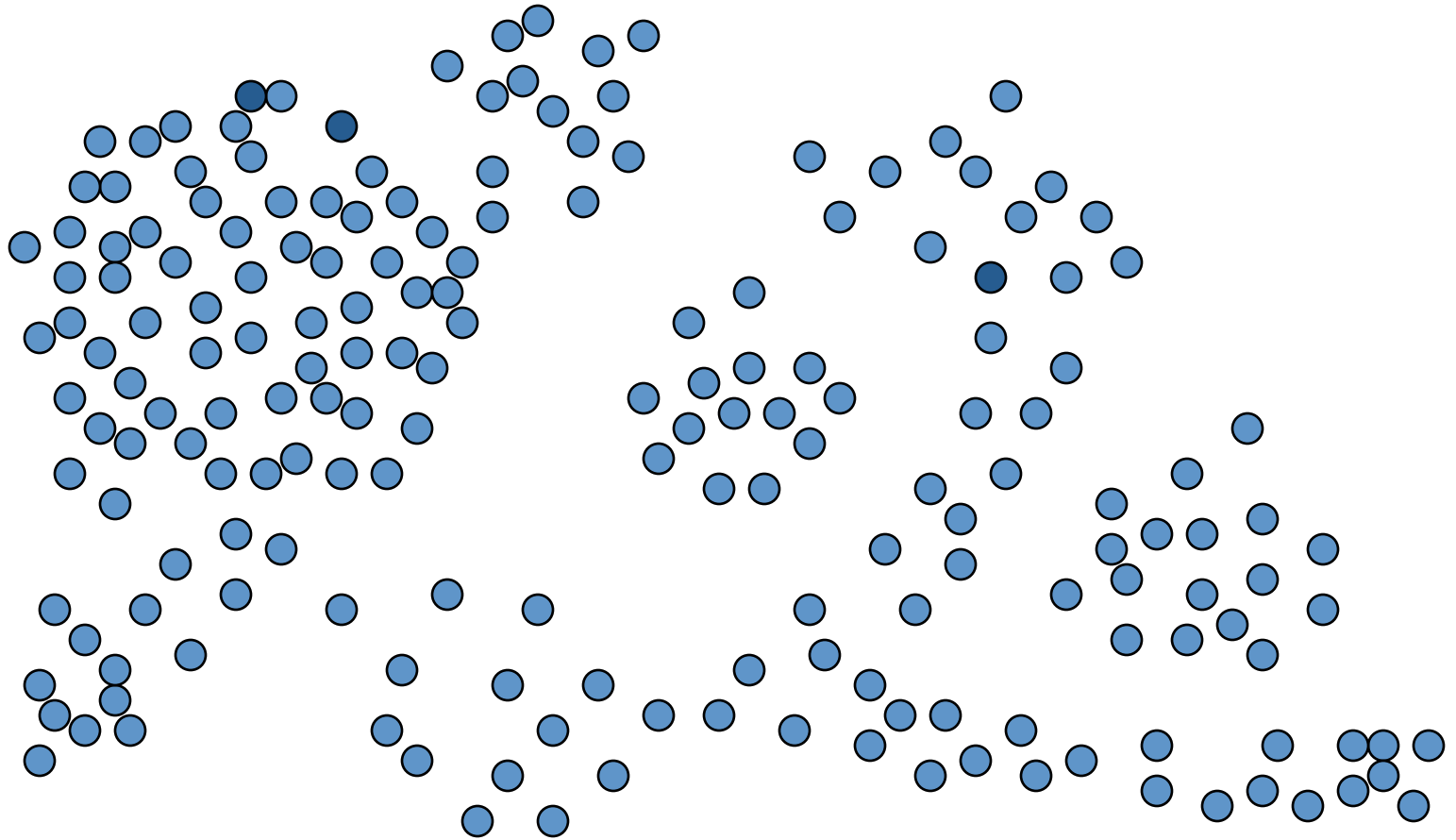
Inne algorytmy grupowania

Wykorzystują inne paradygmaty, np.

- **Gęstość** obiektów w przestrzeni cech
- Strukturę sieci komórek – tzw. **grid**

Skupisko – obszar charakteryzujący się dużą gęstością obiektów. Skupiska obiektów odseparowane od siebie obszarami o małej gęstości występowania obiektów

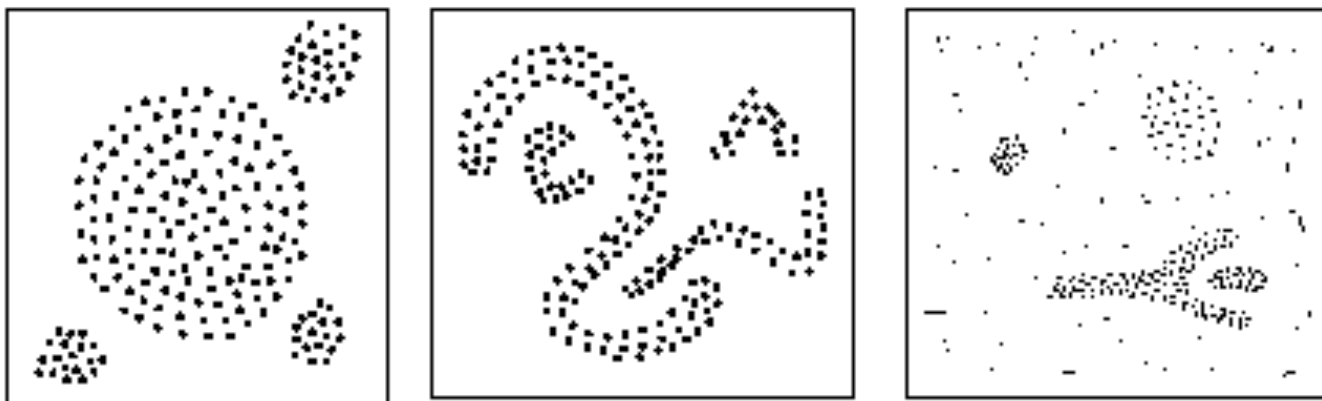
Modelowanie dowolnych kształtów



Metody gęstościowe

- Wykorzystują pojęcie gęstości (ang. density) – **lokalne sąsiedztwo** punktu/skupienia, a także „gęsto” połączonych punktów
- Podstawowy pomysł – przyrostowe tworzenie skupiska poprzez dołączanie obiektów należących do najbliższego sąsiedztwa tego skupiska pod warunkiem, że spełniają pewne minimalne parametry
- Właściwości metod gęstościowych:
 - Wykrywanie skupień o dowolnych kształtach (niesferycznych)
 - Odporność na „szum informacyjny” i obs. „outliers”
 - Samodzielne określanie liczby potrzebnych skupisk
 - Potrzebna parametryzacja oceny gęstości i warunków zatrzymania
- Znane algorytmy:
 - DBSCAN: Ester, et al. (KDD’96)
 - OPTICS: Ankerst, et al (SIGMOD’99).
 - DENCLUE: Hinneburg & D. Keim (KDD’98)
 - CLIQUE: Agrawal, et al. (SIGMOD’98)
 - Liczne rozszerzanie

Metody gęstościowe – ang. Density-Based Clustering



- Skupiska grupują gęste punktu, obszary i są odległe od innych gęstych obszarów lub „rzadkich” punktów
- Jak oceniać gęstość i jakość skupiska?
- DBSCAN - grupowanie wykorzystujące ocenę gęstości rozkładu (lokalne kryterium – sąsiedztwo punktu), parametry oczekiwanej gęstości (minimalna liczba punktów) oraz wielkości sąsiedztwa

DBSCAN: Algorytm gęstościowy

- DBSCAN: Density Based Spatial Clustering of Applications with Noise (Ester et al.'96)
 - Wprowadza pojęcie „*density-based cluster*”: Skupienie będące największym zbiorem punktów gęsto połączonych „*density-connected points*” (ze względu na parametry sąsiedztwa punktów)
 - Skupienie to zbiór obiektów wzajemnie osiągalnych lub połączonych z pewną zadaną gęstością
 - Wykorzystuje ϵ -sąsiedztwo punktu i możliwość podziały obiektów na typy (wg. liczby sąsiadów) oraz zasady przemieszczania się pomiędzy nimi (idea osiągalności)
 - Możliwość wykrywania skupień o dowolnym kształcie w obecności szumu informacyjnego (ang noise) i obserwacji samotniczych

DBSCAN: Podstawowe pojęcia

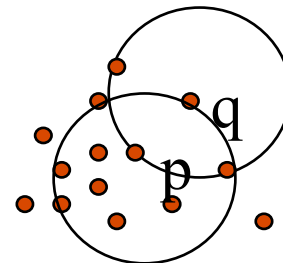
- Parametry:
 - **Eps** (ϵ): Maksymalny promień sąsiedztwa
 - **MinPts**: minimalna liczba punktów (obiektów) w Eps-sąsiedztwie badanego punktu
- D – dany zbiór obiektów do pogrupowania; wybrana miara odległości $dist(p,q)$
- ϵ sąsiedztwo obiektu p to zbiór innych punktów q spełniających
$$N_{Eps}(p) : \{\text{punkt } q \text{ należy do } D \mid dist(p,q) \leq Eps\}$$

W zależności od liczby sąsiadów w otoczeniu obiektu p:

Obiekt rdzenia

Obiekt graniczny

Obiekt oddalony / Szum (ang. noise point)



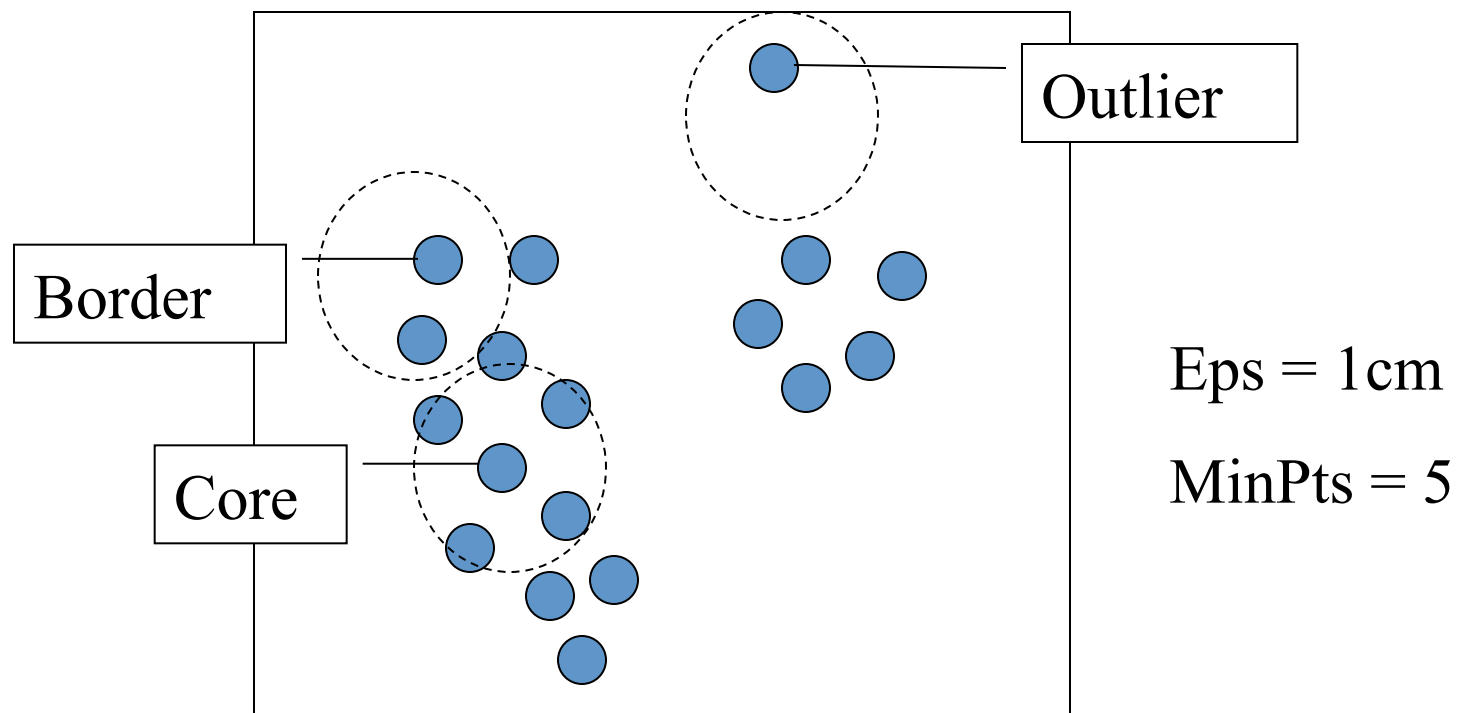
MinPts = 5

Eps = 1 cm

Typy obiektów

- **Obiekt centralny – rdzeń** (ang. core point) = obiekt, który ma co najmniej MinPts sąsiednich obiektów w swoim ϵ sąsiedztwie (są to założki do budowy gęstych skupisk)
- **Obiekt brzegowy** (ang. border point) = obiekt mający mniej niż MinPts sąsiednich obiektów w swoim ϵ sąsiedztwie, lecz należący do sąsiedztwa co najmniej jednego punktu rdzenia
- **Obiekt oddalony / szum** (ang. noise point) = obiekt z liczbą sąsiadów niż MinPts nie należący do sąsiedztwa innych punktów rdzeniowych (odległy o więcej niż ϵ od innych potencjalnych skupisk)

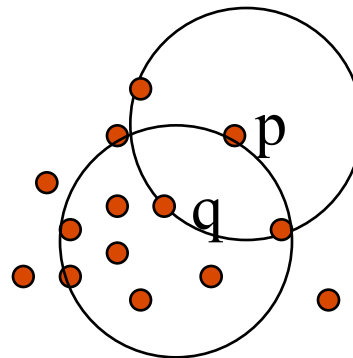
DBSCAN – ilustracja typów obiektów



DBSCAN: Podstawowe pojęcia (2)

- Osiągalność obiektów (niezbędna do tworzenia skupisk)
- $N_{eps}(q)$: {punkt p należy do D / $dist(p,q) \leq Eps$ }
- **Bezpośrednia osiągalność gęstościowa**
- Mówimy, że obiekt p jest bezpośrednio osiągalny z punktu q (ze względu na parametry Eps , $MinPts$), jeśli
 - 1) p należy do ϵ sąsiedztwa $N_{Eps}(q)$
 - 2) Obiekt q jest centralny:

$$|N_{Eps}(q)| \geq MinPts$$

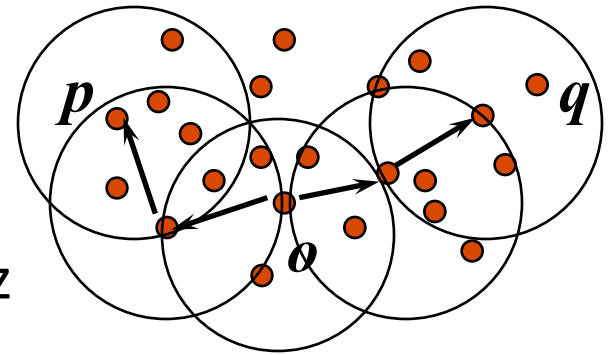
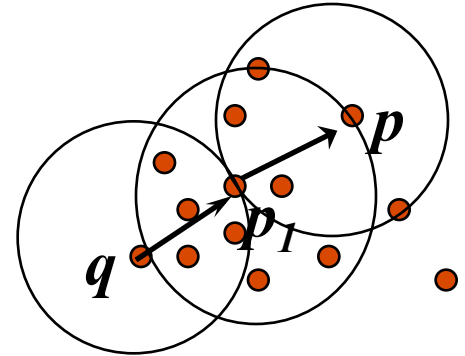


$MinPts = 5$

$Eps = 1 \text{ cm}$

DBSCAN: Podstawowe pojęcia (3)

- Gęstościowa osiągalność (Density-reachable):
 - Obiekt p jest gęstościowo osiągalny z punktu q jeśli istnieje łańcuch punktów pośrednich $p_1, \dots, p_n, p_1 = q, p_n = p$, takich że p_{i+1} jest bezpośrednio osiągalny z p_i
- Połączeniowa gęstości (Density-connected)
 - Obiekt p jest gęstościowo połączony z obiektem q (wrt. $Eps, MinPts$) jeśli istnieje punkty o taki, że obiekty p oraz q są z niego gęstościowo osiągalne
- Połączenia / osiągalność pozwalają na określenie skupiska zaczynając z jednego z obiektów centralnych



DBSCAN: Zarys algorytmu

- Wybierz punkt startowy p
- Odnajdź wszystkie punkty do gęstościowego osiągnięcia z p (density-reachable from p wrt ***Eps*** and ***MinPts***).
- Jeśli p jest rdzeniem (*core point*), utwórz skupienie.
- Jeśli p jest punktem granicznym (border point) i żadne punkty nie są z niego gęstościowo osiągalne, DBSCAN wybiera następny punkt z bazy danych
- Proces jest kontuowany dopóki żaden nowy punkt nie może być dodany to dowolnego skupienia
- Punkty które nie są rdzeniem lub graniczne i nie mogą być zaliczone do skupisk – stają się punktami oddalonymi (noise)
- Złożoność: $O(n \log n)$ w przypadku użycia specjalnego „spatial index”, w przeciwnym razie $O(n^2)$.

DBSCAN trudności parametryzacji

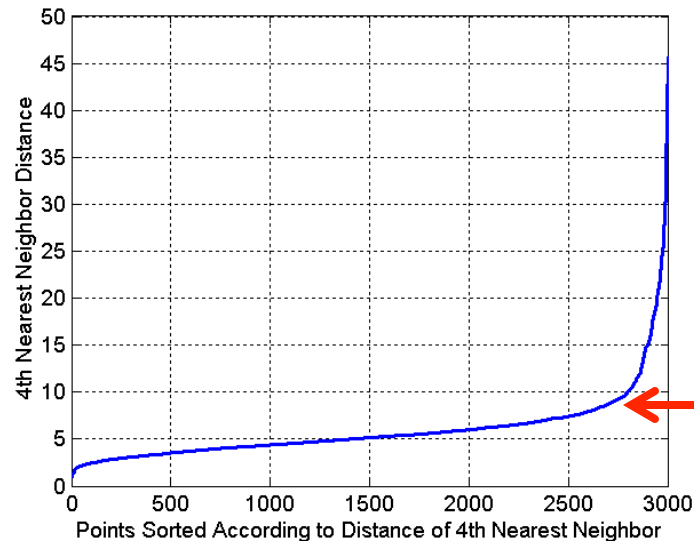
Algorytm jest dość czuły na dobór parametrów Eps (ϵ) i MinPts

Autorzy zaproponowali heurystykę:

- Niech d będzie odległością obiektu p do jego k -tego najbliższego sąsiada, sąsiedztwem obiektu p będzie dokładnie $k + 1$ obiektów
- Należy dobrać k dla danych D
- Określić funkcję k -dist, która odwzorowuje każdy obiekt p w danych D na odległość do jego k -tego najbliższego sąsiada
- Uporządkuj wartości k -dist dla obiektów (wykres gęstości)
 - Dla obiektu p - ustawiając wartość parametru Eps na k -dist(p), a wartość parametru MinPts na k , wszystkie obiekty z mniejszą lub równą wartością k -dist staną się obiektami wewnętrznymi sąsiedztwa / skupiska / inne są kandydatami na punkty oddalone
- Znajdź punkt progowy przegięcia wykresu

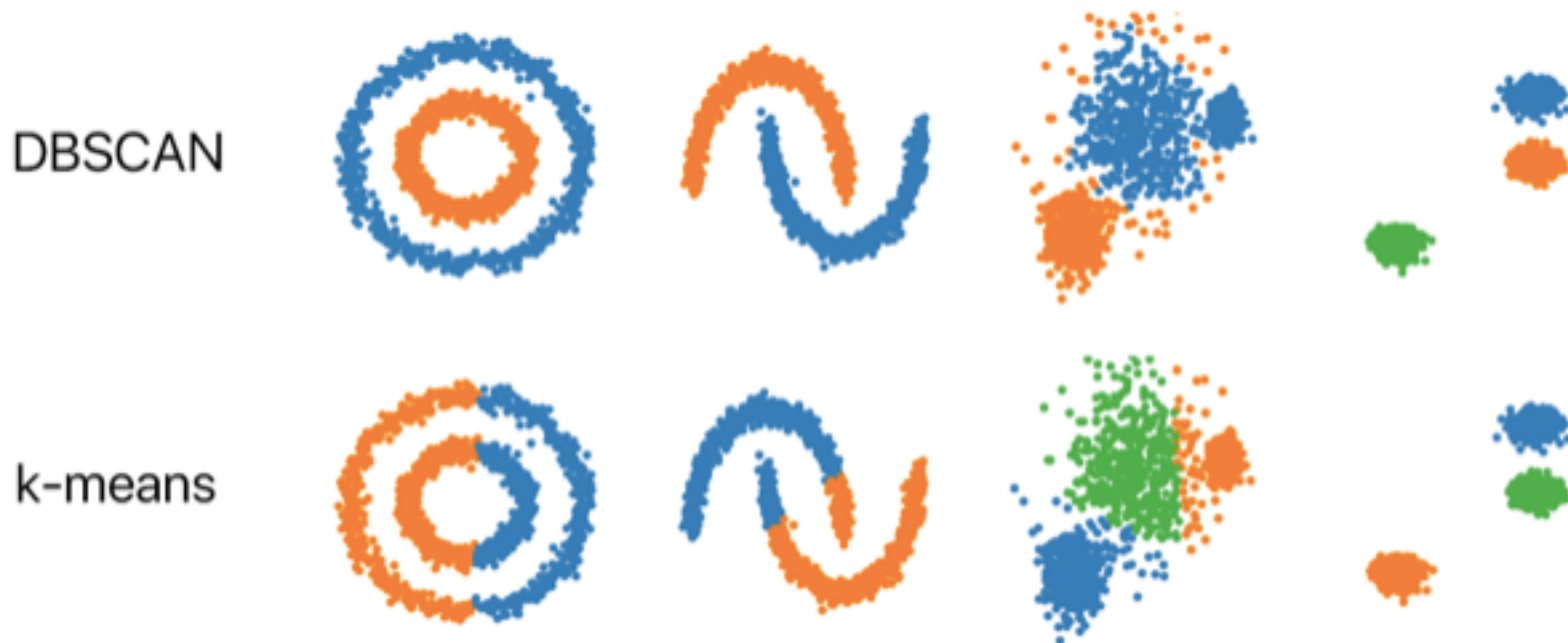
DBSCAN: Wykres k-sasiedztwa

- Punkty w gęstym skupisku, większość ich k^{th} najbliższych sąsiadów ma podobną wartość odległości
- Obiekty oddalone oraz szum (ang. noise points) ich k^{th} najbliższy sąsiad jest dużo bardziej odległy (odległość wyraźnie rośnie)
- Posortuj obiekty wg. ich odległości od k^{th} najbliższego sąsiada i zrób wykres
- Znajdź odległość d odpowiadającej przegięciu kształtu wykresu (ang. “knee” in the curve)
 - $\text{Eps} = d$, $\text{MinPts} = k$



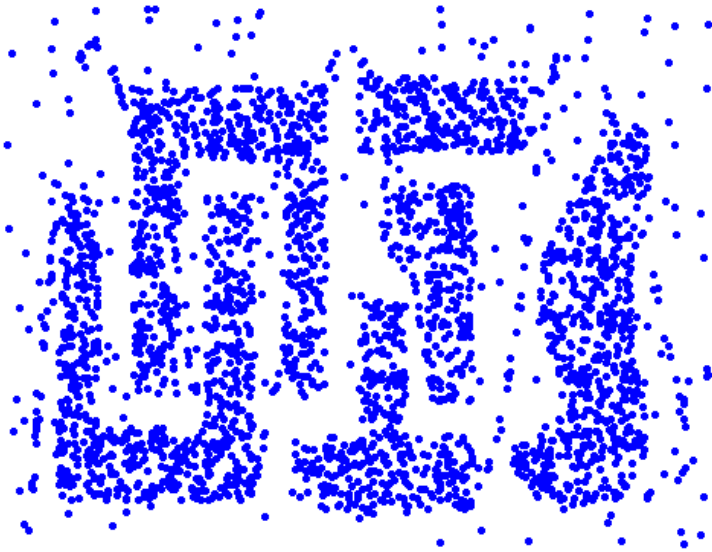
Eps ~ 7-10
MinPts = 4

Przykłady porównania algorytmów

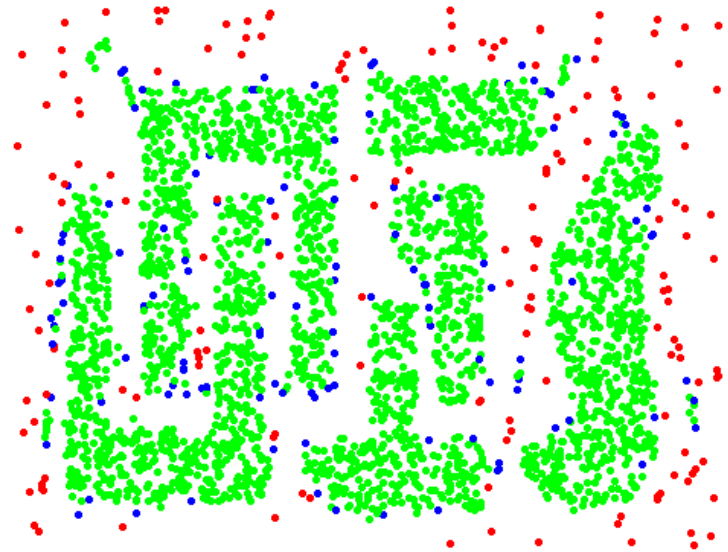


Więcej w blogu pt. An introduction to the DBSCAN algorithm and its Implementation in Python [Nagesh Singh Chauhan] KDnuggets

DBSCAN: przykład użycia



Oryginalne dane



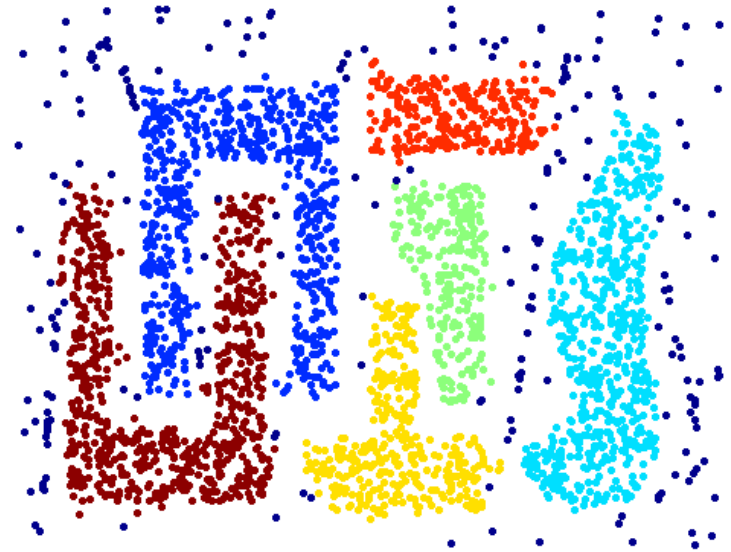
Typy obiektów: **core**,
border and **noise**

Eps = 10, MinPts = 4

Skupiska DBSCAN



Original Points




Clusters

- Radzi sobie z szumem informacyjnym
- Tworzy skupiska o różnych kształtach

Liczne implementacje DBSCAN

Scikit learn - cluster

Także inne języki i środowiska,
WEKA +
DBSCAN. Lightweight Java

 [Install](#) [User Guide](#) [API](#) [Examples](#) [More](#) [Go](#)

[Prev](#) [Up](#) [Next](#)

scikit-learn 0.24.2
[Other versions](#)

Please [cite us](#) if you use the software.

sklearn.cluster.DBSCAN
Examples using **sklearn.cluster.DBSCAN**

sklearn.cluster.DBSCAN

```
class sklearn.cluster.DBSCAN(eps=0.5, *, min_samples=5, metric='euclidean', metric_params=None, algorithm='auto', leaf_size=30, p=None, n_jobs=None)
```

[\[source\]](#)

Perform DBSCAN clustering from vector array or distance matrix.

DBSCAN - Density-Based Spatial Clustering of Applications with Noise. Finds core samples of high density and expands clusters from them. Good for data which contains clusters of similar density.

Read more in the [User Guide](#).

Parameters:

- eps : float, default=0.5**
The maximum distance between two samples for one to be considered as in the neighborhood of the other. This is not a maximum bound on the distances of points within a cluster. This is the most important DBSCAN parameter to choose appropriately for your data set and distance function.
- min_samples : int, default=5**
The number of samples (or total weight) in a neighborhood for a point to be considered as a core point. This includes the point itself.
- metric : string, or callable, default='euclidean'**
The metric to use when calculating distance between instances in a feature array. If metric is a string or callable, it must be one of the options allowed by [sklearn.metrics.pairwise_distances](#) for its metric parameter. If metric is "precomputed", X is assumed to be a distance matrix and must be square. X may be a [Glossary](#), in which case only "nonzero" elements may be considered neighbors for DBSCAN.

[Toggle Menu](#)

Przykład DBSCAN -sklearn

```
1 import numpy as np
2 from sklearn.cluster import DBSCAN
3 from sklearn import metrics
4 from sklearn.datasets import make_blobs
5 from sklearn.preprocessing import StandardScaler
6
7 # Generate sample data
8 centers = [[1, 1], [-1, -1], [1, -1]]
9 X, labels_true = make_blobs(n_samples=750, centers=centers, cluster_std=0.4,
10                             random_state=0)
11
12 X = StandardScaler().fit_transform(X)
13
14 # Compute DBSCAN
15 db = DBSCAN(eps=0.3, min_samples=10).fit(X)
16 core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
17 core_samples_mask[db.core_sample_indices_] = True
18 labels = db.labels_
19
20 # Number of clusters in labels, ignoring noise if present.
21 n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
```

Grupowanie z wykorzystaniem modeli prawdopodobieństwa

- Podejścia oparte na założeniu, że dane są generowane w wyniku realizacji pewnego procesu statystycznego
- Zakłada się pewien model rozkładu prawdopodobieństwa występowanie obserwacji
- Każdemu potencjalnemu skupisku odpowiada model, w postępowaniu (algorytmie) weryfikuje się stopień dobrego dopasowania oryginalnych danych do przyjętego modelu
- Celem grupowania jest znalezienie zbioru (mieszaniny) modeli opisujących skupiska oraz estymacja parametrów tych modeli
- Obiekty przydziela się do skupisk zgodnie ze sparametryzowanymi modelami i zasadą klasyfikacji Bayesowskiej
- Patrz kolejny wykład

Pytanie i komentarze?

Dalszy kontakt:

jerzy.stefanowski@cs.put.poznan.pl

<http://www.cs.put.poznan.pl/jstefanowski/>



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego

